Qualitative Symbolic Perturbation: Two Applications of a New Geometry-based Perturbation Framework

Olivier Devillers, Menelaos Karavelas, Monique Teillaud

QUALITATIVE SYMBOLIC PERTURBATION: TWO APPLICATIONS OF A NEW GEOMETRY-BASED PERTURBATION FRAMEWORK *

Olivier Devillers[†‡§] *Menelaos I. Karavelas*[¶‖] *Monique Teillaud*[†‡§]

Abstract. In a classical *Symbolic Perturbation* scheme, degeneracies are handled by substituting some polynomials in ε for the inputs of a predicate. Instead of a single perturbation, we propose to use a sequence of (simpler) perturbations. Moreover, we look at their effects geometrically instead of algebraically; this allows us to tackle cases that were not tractable with the classical algebraic approach.

1 Introduction

In earlier computational geometry papers, the treatment of degenerate configurations was mainly ignored. However, degenerate situations actually do occur in practice. When data are highly degenerate by nature, a direct handling of special cases in a particular algorithm can be efficient [5]. But in many situations, degeneracies happen only occasionally, and perturbation schemes are an easy and efficient generic solution. Controlled perturbations [18] combine increasing arithmetic precision together with actual displacement of the data, and eventually compute a non-degenerate configuration. On the other hand, the use of a symbolic perturbation allows a geometric algorithm or data structure that was originally designed without addressing degeneracies, to still operate on degenerate cases, without concretely modifying the input [10, 21, 22]. Actually, similar strategies were often used by earlier implementors of simple geometric algorithms, without identifying them as symbolic perturbations: for instance when incrementally computing a convex hull, when the new inserted point was lying on a facet of the convex hull, the point was decided to be inside the convex hull.

Let us formalize the notions of geometric problem and degeneracy. Let $G(\boldsymbol{u})$ be a geometric structure depending on some input data $\boldsymbol{u}$. The input space $\mathcal{I}$ to which the input $\boldsymbol{u}$ belongs is usually some space with a natural topology; e.g., if $\boldsymbol{u}$ is a set of n points in

*The work in this paper has been partially supported by the FP7-REGPOT-2009-1 project "Archimedes Center for Modeling, Analysis and Computation". M. Karavelas acknowledges support by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) – Research Funding Program: THALIS – UOA (MIS 375891).

† Inria, Centre de recherche Nancy - Grand Est, France.
members.loria.fr/Olivier.Devillers,members.loria.fr/Monique.Teillaud

‡ CNRS, Loria, France.

§ Université de Lorraine, France.

¶ Mathematics and Applied Mathematics Dept., U. Crete, Greece. www.tem.uoc.gr/∼mkaravel/

‖ Institute of Applied and Computational Mathematics, FORTH, Greece.

the plane, the input space is $\mathcal{I} = \mathbb{R}^{2n}$. The output space $\mathcal{O}$ may be more fuzzy since the structure can often be described in various ways; for example, the convex hull of $\boldsymbol{u} \in \mathbb{R}^{2n}$ can be defined as a (compact convex) subset of $\mathbb{R}^2$, then $\mathcal{O} \subset 2^{\mathbb{R}^2}$, or as the finite subset of $\mathbb{N}$ containing the indices of the extreme vertices, then $\mathcal{O} = 2^{\mathbb{N}}$. We choose a combinatorial definition of the geometric structure so that the output space is countable; for instance we define a planar Voronoi diagram as the set of triples of indices of points defining Voronoi vertices, i.e., a finite subset of $\mathbb{N}^3$: $\mathcal{O} = 2^{\mathbb{N}^3}$. For such a discrete space the usual topology is the trivial topology, i.e., the singleton $\{x\}$ is a valid neighborhood of point x. In this setting, a degenerate configuration $\boldsymbol{u}_0$ for the geometric structure G is an input position $\boldsymbol{u}_0$ where G is not continuous, i.e., not constant since we use the trivial topology on $\mathcal{O}$; for a degenerate configuration $\boldsymbol{u}_0$, the geometric structure $G(\boldsymbol{u_0})$ may be undefined (or not uniquely defined).

A *symbolic perturbation* consists in using as input $\boldsymbol{u}$ for G the values of a continuous function $\pi\colon \mathcal{I} \times \mathbb{R}_{\geq 0} \to \mathcal{I}$ of a parameter ε. This is done in such a way that for $\varepsilon = 0$ the value of $\pi(\boldsymbol{u}_0, 0)$ is equal to $\boldsymbol{u}_0$, and $\pi(\boldsymbol{u}_0, \varepsilon)$ is non-degenerate for G for sufficiently small positive values of ε. We define $\hat{G}$ as $\hat{G}(\boldsymbol{u}) = \lim_{\varepsilon \to 0^+} G(\pi(\boldsymbol{u}, \varepsilon))$. If $\boldsymbol{u}_0$ is a non-degenerate configuration, i.e., if G is continuous in $\boldsymbol{u_0}$, then by continuity $\hat{G}(\boldsymbol{u}_0) = G(\boldsymbol{u}_0)$. Otherwise, $G(\boldsymbol{u}_0)$ may be either undefined or different from $\hat{G}(\boldsymbol{u}_0)$. With a slight abuse of notation, in the rest of this paper we still denote the modified function $\hat{G}$ as G.

A symbolic perturbation allows an algorithm that computes $G(\boldsymbol{u})$ in generic situations to compute $G(\boldsymbol{u}_0)$ for the degenerate input $\boldsymbol{u}_0$. Most decisions made by the algorithm are usually made by looking at *geometric predicates*, which are combinations of elementary predicates. An elementary predicate is the sign of a continuous real function of the input, where the output space for the sign function is $\{-1, 0, 1\}$. A configuration is algorithmically degenerate, when some predicate used by the algorithm returns 0. When applying a symbolic perturbation, a predicate $\operatorname{sign}(p(\boldsymbol{u}))$ evaluated at $\boldsymbol{u}_0$ returns the limit of $\operatorname{sign}(p(\pi(\boldsymbol{u}_0, \varepsilon)))$ as $\varepsilon \to 0^+$. A perturbation scheme is said to be *effective* for a predicate $\operatorname{sign}(p(\boldsymbol{u}))$ if for any $\boldsymbol{u}_0$ the limit exists, and it is non-zero. The perturbed sign of $p(\boldsymbol{u}_0)$ is then given by this limit.

The main difficulty when designing a perturbation scheme for $G(\boldsymbol{u})$ is to find a function $\pi(\boldsymbol{u}_0, \varepsilon)$, such that the perturbation scheme can be proved to be effective for all relevant functions $p(\boldsymbol{u})$, and the perturbed predicates are easy to evaluate, e.g., using as few as possible arithmetic operations. The work of designing and proving the effectiveness of a perturbation for G is typically tailored to a specific algorithm for computing the geometric structure G.

In previous works [1, 9, 10, 11, 19], a predicate is the sign of a polynomial P in some input $\boldsymbol{u} \in \mathbb{R}^m$. The input $\boldsymbol{u}$ is perturbed as an element $\pi(\boldsymbol{u}_0, \varepsilon)$ of $\mathbb{R}^m$ whose coordinates are polynomials in $\boldsymbol{u}_0$ and ε, such that $\pi(\boldsymbol{u}_0, \varepsilon)$ goes to $\boldsymbol{u}_0$ when $\varepsilon \to 0^+$. The perturbed predicate returns the sign of the limit $\lim_{\varepsilon \to 0^+} \operatorname{sign}(P(\pi(\boldsymbol{u}_0, \varepsilon)))$. Since P is a polynomial, $P(\pi(\boldsymbol{u}_0, \varepsilon))$ can be rewritten as a polynomial in ε whose monomials in ε are ordered in terms of increasing degree. The constant monomial is actually $P(\boldsymbol{u}_0)$, while the signs of the remaining coefficients can be viewed as auxiliary predicates on $\boldsymbol{u}_0$. The coefficients of $P(\pi(\boldsymbol{u}_0, \varepsilon))$ are evaluated in increasing degrees in ε, until a non-vanishing coefficient is found. The sign of this coefficient is then returned as the value of the predicate $\text{sign}(P(\boldsymbol{u}_0))$.

Contribution In this paper we propose QSP (Qualitative Symbolic Perturbation), a new framework for resolving degenerate configurations in geometric computing. Unlike classical symbolic perturbation techniques, QSP resolves degeneracies in a purely geometric manner, and independently of a specific algebraic formulation of the predicate. So, the technique is particularly suitable for predicates whose algebraic description is not unique or too complicated, such as the ones treated in this paper. In fact, QSP can even handle predicates that are signs of non-polynomial functions.

In addition, instead of having a single perturbation parameter that governs the way the input objects or predicates are modified, QSP allows for a sequence of perturbation parameters: conceptually, we symbolically apply the different perturbations one after the other, using a well-defined canonical ordering that corresponds to considering first the biggest perturbation. To achieve termination, we must devise an appropriate sequence of perturbations which guarantees that eventually, i.e., after having perturbed sufficiently many input objects, the degenerate predicate is resolved in a non-degenerate manner. The number of objects that need to be perturbed depends on the specific predicate that we analyze. For example in the 2D Apollonius diagram, for a given predicate, perturbing a single object (among the objects involved in the predicate evaluation) always suffices, whereas in its 3D counterpart, we may need to perturb two input objects (among the objects involved in the predicate evaluation).

For a predicate expressed as the sign of a polynomial, the arithmetic degree of a predicate is the minimal degree of a polynomial (or of a combination of a constant number of polynomials) that allows to answer the predicate; the degree of the polynomial is a rough measure of the precision needed to evaluate the predicate exactly and of the error introduced by floating point computation [2, 6, 17]. Standard algebraic symbolic perturbation schemes [10, 11, 19] automatically provide the auxiliary predicates that need to be evaluated. The corresponding auxiliary polynomials are, by design, of at most the same algebraic degree as the original polynomial, but evaluating their sign in an efficient manner (e.g., by factorizing) is far from being an obvious task. QSP schemes cannot guarantee that the auxiliary polynomials are of lower algebraic degree than the original one; however, in principle, the auxiliary predicates that we have to deal with are expected to be more tractable, since their analysis is based on geometric considerations.

As for any symbolic perturbation scheme, QSP assumes exact arithmetic to detect degeneracies. For many applications, degeneracies are rare enough to allow for high efficiency using the exact geometric computation paradigm [23].

In the next section of the paper we formally define the QSP framework. In Section 3 and 4 we describe QSP schemes for the main predicates of two geometric structures: (1) the 2D arrangement of circular arcs and (2) the 2D/3D Apollonius diagram. We end with Section 5, where we discuss the advantages and disadvantages of our framework, and indicate directions for future research.

2 General framework

Let us start with a simple example. Assume that the input is a set of points $q_0, q_1 \dots q_{n-1}$ to be sorted by x-coordinates. Degeneracies occur when several points have the same abcissa. A possible perturbation consists in translating point q_i horizontally to the right by ε_i, for any i, while ensuring that $\varepsilon_0 \lll \varepsilon_1 \lll \ldots \lll \varepsilon_{n-1}$. Then, when comparing the abcissae of two points with the same abcissa, the one that is considered the rightmost is the one perturbed by the largest translation, which is the one with highest index. If the points are numbered by increasing y-coordinates, then sorting them by perturbed x-coordinates reduces to a lexicographic sorting, which is the usual way of solving equal abcissae degeneracies.

QSP formalizes this approach. The next section formally describes QSP and Section 2.2 illustrates its behavior with simple toy examples.

2.1 The QSP scheme

Let $G(\boldsymbol{u})$ be a geometric structure whose computation depends on a predicate $\mathrm{sign}(p(\boldsymbol{u}))$, where p is a continuous real function. In this formal presentation, p appears as a function of the whole input $\boldsymbol{u}$; however, in practice, a predicate depends only on a constant size subset of $\boldsymbol{u}$.

We design the perturbation scheme π as a sequence of successive symbolic perturbations π_i (as defined in introduction), $0 \le i < N$:

$$\pi(\boldsymbol{u}, \boldsymbol{\varepsilon}) = \pi_0(\pi_1(\pi_2(\ldots \pi_{N-1}(\boldsymbol{u}, \varepsilon_{N-1}) \ldots, \varepsilon_2), \varepsilon_1), \varepsilon_0),$$

where $\boldsymbol{\varepsilon} = (\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{N-1}) \in \mathbb{R}^N_{\ge 0}$. The number of perturbations N is part of the perturbation scheme and usually depends on the input size. Since $\boldsymbol{\varepsilon}$ is no longer a single real number, we have to determine how the limit is taken; we thus define $G(\boldsymbol{u})$ to be the limit:

$$G(\boldsymbol{u}) = \lim_{\varepsilon_{N-1} \to 0^+} \lim_{\varepsilon_{N-2} \to 0^+} \cdots \lim_{\varepsilon_1 \to 0^+} \lim_{\varepsilon_0 \to 0^+} G(\pi(\boldsymbol{u}, \boldsymbol{\varepsilon})).$$

Note that writing the limits in this specific order implies that $\varepsilon_0 \lll \varepsilon_1 \lll \ldots \lll \varepsilon_{N-1}$. In other words, perturbation π_i is bigger than perturbation π_{i-1}, for $i = N-1, \ldots, 1$.

Each perturbed predicate $\mathrm{sign}(p(\boldsymbol{u}))$ is evaluated as

$$\lim_{\varepsilon_{N-1} \to 0^+} \lim_{\varepsilon_{N-2} \to 0^+} \cdots \lim_{\varepsilon_1 \to 0^+} \lim_{\varepsilon_0 \to 0^+} \mathrm{sign}(p(\pi(\boldsymbol{u}, \boldsymbol{\varepsilon}))). \tag{1}$$

As for standard symbolic perturbations, the perturbation is effective if this limit always exists and is non-zero.

In this definition, the order of evaluation of the limits is standard, i.e., the limit when ε_i goes to 0 is taken before the limit when ε_{i+1} goes to 0, for $i = 0, \ldots, N-2$. The following theorem states that, under some conditions, it is possible to evaluate the limits in an easier way.

Theorem 1. *Let* $\mathrm{sign}(p(\boldsymbol{u}))$ *be a predicate and let* π *be a sequence of perturbations as above. We assume the existence of the following limits, for* $0 < i \le N$,

$$\begin{aligned}\ell_i &= \lim_{\varepsilon_{N-1}\to 0^+}\lim_{\varepsilon_{N-2}\to 0^+}\dots\lim_{\varepsilon_{N-i}\to 0^+}\mathrm{sign}(p(\pi(\boldsymbol{u},(0,0,\dots,0,\varepsilon_{N-i},\varepsilon_{N-i+1},\dots,\varepsilon_{N-1})))),\\ \ell'_i &= \lim_{\varepsilon_{N-i}\to 0^+}\mathrm{sign}(p(\pi(\boldsymbol{u},(0,0,\dots,0,\varepsilon_{N-i},0,\dots,0)))),\end{aligned}$$

With this notation, the value of the perturbed predicate defined in (1) is ℓ_N.

If the perturbation is effective, i.e., $\ell_N \neq 0$, *let* μ *be the smallest index* i *for which* ℓ_i *is non-zero. Then the value of the perturbed predicate is* $\ell_N = \ell_\mu$.

Furthermore, if $\mathrm{sign}(p(\pi(\boldsymbol{u},(0,0,\dots,0,\varepsilon_{N-\mu},\varepsilon_{N-\mu+1},\dots,\varepsilon_{N-1}))))$ *does not depend on* $\varepsilon_{N-\mu},\varepsilon_{N-\mu+1},\dots,\varepsilon_{N-1}$ *in a neighborhood of the origin in* $\mathbb{R}^{\mu-1}_{\ge 0}$, *except at the origin, then* $\ell_N = \ell_\mu = \ell'_\mu$.

The proof of the theorem relies on two lemmas about the limit of the sign of a function of two variables, which formalize two trivial observations.

Lemma 2. *Let* f *be a continuous function of two variables* (a,b) *defined in a neighborhood of the origin. If defined, let* $\lim_{b\to 0^+}\mathrm{sign} f(0,b)$ *be denoted as* s.

$$\text{If } s \neq 0 \text{ then } \lim_{b\to 0^+}\lim_{a\to 0^+}\mathrm{sign} f(a,b) = s.$$

Proof. Let us assume that $s \neq 0$, i.e., $s \in \{-1,+1\}$. Then there exists $\delta > 0$ such that $\forall b \in (0,\delta]$, $f(0,b)$ has constant sign s (thus it does not vanish). For any b fixed in $(0,\delta]$ the function $f(a,b)$ is a continuous function in variable a, thus $\lim_{a\to 0^+} f(a,b) = f(0,b)$. Since $f(0,b) \neq 0$, the sign of $f(a,b)$ is the same as the sign of $f(0,b)$ when a is in a neighborhood of 0, so $\lim_{a\to 0^+}\mathrm{sign} f(a,b) = \mathrm{sign} f(0,b) = s$. Thus $\lim_{b\to 0^+}\lim_{a\to 0^+}\mathrm{sign} f(a,b) = \lim_{b\to 0^+} s = s$. □

The function $f(a,b) = a$ shows that the hypothesis $s \neq 0$ is necessary.

Lemma 3. *Let* f *be a continuous function in two variables* (a,b) *defined in a neighborhood* $U \times U$ *of the origin. Assume that* $\forall a \in U$, $\forall b,b' \in U$ *with* $a > 0$ *and* $b,b' \ge 0$ *we have* $\mathrm{sign} f(a,b) = \mathrm{sign} f(a,b')$ *and* $\lim_{a\to 0^+}\mathrm{sign} f(a,0)$ *exists. Then*

$$\lim_{b\to 0^+}\lim_{a\to 0^+}\mathrm{sign} f(a,b) = \lim_{a\to 0^+}\mathrm{sign} f(a,0).$$

Proof. Let $s = \lim_{a\to 0^+}\mathrm{sign} f(a,0)$. There exists $\delta > 0$ such that $\forall a \in (0,\delta]$, $\mathrm{sign} f(a,0) = s$. By the hypothesis in the lemma we have $\forall a \in (0,\delta]$, $\forall b \ge 0$, $\mathrm{sign} f(a,0) = \mathrm{sign} f(a,b) = s$. The function has constant sign s on $((0,\delta] \times (0,\infty)) \cap (U \times U)$ so the limit is s. □

Proof of Theorem 1. Applying Lemma 2 to variables $\varepsilon_{N-\mu}$ and $\varepsilon_{N-\mu-1}$ we get $\ell_\mu = \ell_{\mu+1}$ and by induction $\ell_\mu = \ell_{\mu+1} = \ell_{\mu+2} = \ldots = \ell_N$.

Applying Lemma 3 μ times to variables $\varepsilon_{N-\mu}$ and ε_{N-i} for all $i < \mu$ yields $\ell_\mu = \ell'_\mu$ under the hypothesis. □

The strategy of evaluation of the perturbed predicate is the following. We first compute $p(\pi(\boldsymbol{u},(0,0,\ldots,0)))=p(\boldsymbol{u})$; if non-zero, its sign is the result for the predicate, i.e., the value of the perturbed predicate is the value of the non-perturbed predicate in non-degenerate configurations.

If $p(\boldsymbol{u})=0$, we look at the function $p(\pi(\boldsymbol{u},(0,0,\ldots,\varepsilon_{N-1})))=p(\pi_{N-1}(\boldsymbol{u},\varepsilon_{N-1}))$; if this function is not vanishing when ε_{N-1} lies in a sufficiently small neighborhood to the right of 0, its sign can be returned. In other words, the biggest perturbation π_{N-1} removes the degeneracy; in such a case, smaller perturbations have no influence on the result. More formally, we compute the limit

$$\ell_1=\lim_{\varepsilon_{N-1}\to 0^+}\operatorname{sign}(p(\pi_{N-1}(\boldsymbol{u},\varepsilon_{N-1}))). \tag{2}$$

If ℓ_1 is non-zero, using Theorem 1, it is returned as the value of the predicate $\operatorname{sign}(p(\boldsymbol{u}))$. Otherwise, we have to further perturb our geometric input; we examine the limit

$$\ell_2=\lim_{\varepsilon_{N-1}\to 0^+}\lim_{\varepsilon_{N-2}\to 0^+}\operatorname{sign}(p(\pi_{N-2}(\pi_{N-1}(\boldsymbol{u},\varepsilon_{N-1}),\varepsilon_{N-2}))). \tag{3}$$

If $\ell_2\neq 0$, then the two biggest perturbations π_{N-1} and π_{N-2} actually remove the degeneracy and smaller perturbations are useless. The expression in Eq. (3) can be simplified in cases that actually often occur in applications: if $p(\pi_{N-1}(\boldsymbol{u},\varepsilon_{N-1}))$ is zero on $[0,\eta)$, it is often because the sign of $p(\pi_{N-2}(\pi_{N-1}(\boldsymbol{u},\varepsilon_{N-1}),\varepsilon_{N-2}))$ does not depend on ε_{N-1} in $[0,\eta)$. In such a case, and provided that this function is also non-zero, we can evaluate its sign using the second part of Theorem 1: by taking $\varepsilon_{N-1}=0$, Eq. (3) boils down to

$$\ell_2=\ell_2'=\lim_{\varepsilon_{N-2}\to 0^+}\operatorname{sign}(p(\pi_{N-2}(\boldsymbol{u},\varepsilon_{N-2}))). \tag{4}$$

The process is iterated until a non-zero limit is found. To assert that the perturbation scheme π is effective, we need to prove that one of these limits, ℓ_μ, is indeed non-zero.

When the predicate is a polynomial, we get a sequence of successive evaluations as in algebraic symbolic perturbations; however, the expressions that need to be evaluated have been obtained in a different way and are *a priori* different. The main advantage of this approach is that we may use a very simple perturbation π_ν, since we do not need each perturbation π_ν to be effective, but rather the composed perturbation π. For geometric problems, the simplicity of π_ν allows us to look at the limit in a geometric manner, instead of algebraically computing some appropriate coefficient of $p(\pi(\boldsymbol{u},\varepsilon))$.

2.2 Toy examples

We illustrate these principles with four toy examples. Rather than focusing on one or more geometric structures to be computed, we concentrate on the evaluation of a few predicates using QSP. The examples illustrate how the formal QSP framework, presented in the previous section, arrives at resolving the predicates considered in each example. The formal evaluation yields, as expected, exactly the same result to which we would have arrived at via straightforward geometric arguments.

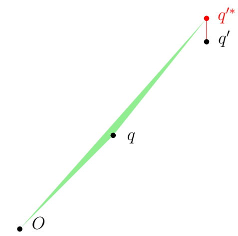
In all examples, we set $\boldsymbol{u} = (q, q') = ((x_0, x_1), (x_2, x_3))$, a pair of two 2D points and $\pi_i(\boldsymbol{u}, \varepsilon_i) = \boldsymbol{u} + \varepsilon_i \boldsymbol{e}_i$, where $\boldsymbol{e}_0 = ((1,0),(0,0))$, $\boldsymbol{e}_1 = ((0,1),(0,0))$, $\boldsymbol{e}_2 = ((0,0),(1,0))$, and $\boldsymbol{e}_3 = ((0,0),(0,1))$ form the canonical basis of $(\mathbb{R}^2)^2$. The differences between the examples below lie in the evaluated predicate $\operatorname{sign}(p(\boldsymbol{u}))$ and the degenerate position $\boldsymbol{u}_0$.

Geometrically, we perturb the input points by first moving their y-coordinates in the positive y-direction, and then their x-coordinates in the positive x-direction. Point q' is considered to be perturbed more than point q. In all examples, the degeneracy is resolved by considering only the perturbed point q'. In some examples perturbing q' along the y-axis suffices to resolve the degeneracy, whereas in others both coordinates of q' need to be perturbed to achieve predicate resolution.

As already mentioned, below we perform an algebraic analysis for each predicate to illustrate the principles of our approach. Notice, however, that the predicate's result for each degenerate configuration is immediate if we consider the perturbations geometrically.

First example: orientation of a flat triangle

Let $p(\boldsymbol{u}) = x_0x_3 - x_1x_2$ and $\boldsymbol{u}_0 = (q, q') = ((1,1),(2,2))$. QSP defines the result for $\operatorname{sign}(p(\boldsymbol{u}_0))$, the orientation of Oqq', as



$$\operatorname{sign}(p(\boldsymbol{u}_0)) = \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\lim_{\varepsilon_0\to 0^+} \operatorname{sign}((1+\varepsilon_0)(2+\varepsilon_3) - (2+\varepsilon_1)(1+\varepsilon_2)).$$

A standard evaluation of this expression would consist in taking the limits in order:

$$\begin{aligned}\operatorname{sign}(p(\boldsymbol{u}_0)) &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+} \operatorname{sign}((2+\varepsilon_3) - (2+\varepsilon_1)(1+\varepsilon_2)) \\ &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+} \operatorname{sign}((2+\varepsilon_3) - 2(1+\varepsilon_2)) \\ &= \lim_{\varepsilon_3\to 0^+} \operatorname{sign}(\varepsilon_3) = 1.\end{aligned}$$

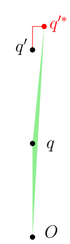
Following the QSP evaluation strategy instead, in such a case, the biggest perturbation, i.e., the perturbation on x_3, allows us to quickly conclude. The only computed limit is the one in Eq. (2):

$$\ell_1 = \lim_{\varepsilon_3\to 0^+} \operatorname{sign}(p((1,1),(2,2+\varepsilon_3))) = \lim_{\varepsilon_3\to 0^+} \operatorname{sign}((2+\varepsilon_3)-2) = \lim_{\varepsilon_3\to 0^+} \operatorname{sign}(\varepsilon_3) = 1.$$

The geometric interpretation is that we get the orientation of a triangle Oqq'^* for a point q'^* slightly above q'.

Second example: orientation of a vertical flat triangle

Let $p(\boldsymbol{u}) = x_0x_3 - x_1x_2$ and $\boldsymbol{u}_0 = (q, q') = ((0,1),(0,2))$. QSP defines the result for $\operatorname{sign}(p(\boldsymbol{u}_0))$, the orientation of Oqq', as



$$\operatorname{sign}(p(\boldsymbol{u}_0)) = \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\lim_{\varepsilon_0\to 0^+} \operatorname{sign}((0+\varepsilon_0)(2+\varepsilon_3) - (1+\varepsilon_1)(0+\varepsilon_2)).$$

Taking the limits in order leads to:

$$\begin{aligned}\operatorname{sign}(p(\boldsymbol{u}_0)) &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\operatorname{sign}(-(1+\varepsilon_1)\varepsilon_2)\\ &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\operatorname{sign}(-\varepsilon_2))\\ &= \lim_{\varepsilon_3\to 0^+}\operatorname{sign}(-1) = -1.\end{aligned}$$

In this case, the QSP evaluation strategy is to first compute

$$\ell_1 = \lim_{\varepsilon_3\to 0^+}\operatorname{sign}(p((0,1),(0,2+\varepsilon_3))) = \lim_{\varepsilon_3\to 0^+} 0 = 0,$$

which does not allow us to resolve the degeneracy. Then we observe that

$$\operatorname{sign}(p((0,1),(x_2,x_3))) = \operatorname{sign}(-x_2)$$

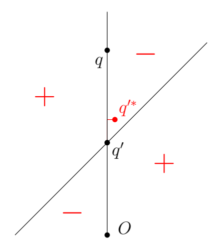
does not depend on x_3, thus we can evaluate ℓ_2 using Eq. (4):

$$\ell_2 = \lim_{\varepsilon_2\to 0^+}\operatorname{sign}(p((0,1),(\varepsilon_2,2))) = \lim_{\varepsilon_2\to 0^+}\operatorname{sign}(-\varepsilon_2) = -1.$$

Two perturbations π_3 and π_2 must be used, but the simplified evaluation of ℓ_2 suffices.

The geometric interpretation is that we look at the orientation of a triangle Oqq'^* for a moved point q'^*. Since moving q'^* slightly above q' doesn't change anything to the degeneracy, the point is moved to the right, which resolves the degeneracy.

Third example: points and quadratic form



Let $p(\boldsymbol{u}) = x_0(x_1-1) - x_0^2 - x_2(x_3-1) + x_2^2$ and $\boldsymbol{u}_0 = (q,q') = ((0,2),(0,1))$.
The predicate p stands for the difference of a degenerate quadratic form evaluated at q and q'. QSP defines the result for $\operatorname{sign}(p(\boldsymbol{u}_0))$ as

$$\operatorname{sign}(p(\boldsymbol{u}_0)) = \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\lim_{\varepsilon_0\to 0^+}\operatorname{sign}\left(\varepsilon_0(2+\varepsilon_1-1) - \varepsilon_0^2 - \varepsilon_2(1+\varepsilon_3-1) + \varepsilon_2^2\right),$$

which could be evaluated as follows:

$$\begin{aligned}\operatorname{sign}(p(\boldsymbol{u}_0)) &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\operatorname{sign}\left(-\varepsilon_2\varepsilon_3 + \varepsilon_2^2\right)\\ &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\operatorname{sign}\left(\varepsilon_2(\varepsilon_2-\varepsilon_3)\right)\\ &= \lim_{\varepsilon_3\to 0^+}\operatorname{sign}(-\varepsilon_3) = -1.\end{aligned}$$

Again the evaluation strategy first computes

$$\ell_1 = \lim_{\varepsilon_3\to 0^+}\operatorname{sign}(p((0,2),(0,1+\varepsilon_3))) = \lim_{\varepsilon_3\to 0^+} 0 = 0,$$

which does not allow us to resolve the degeneracy.

Then we observe that $\mathrm{sign}(p((0,2),(x_2,x_3))) = \mathrm{sign}(x_2(x_3-1)+x_2^2)$ actually depends on x_3, thus we must evaluate ℓ_2 using Eq. (3):

$$\begin{aligned}\ell_2 &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+} \mathrm{sign}(p((0,2),(\varepsilon_2, 1+\varepsilon_3)))\\ &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+} \mathrm{sign}\left(\varepsilon_2(\varepsilon_2-\varepsilon_3)\right)\\ &= \lim_{\varepsilon_3\to 0^+} \mathrm{sign}(-\varepsilon_3) = -1.\end{aligned}$$

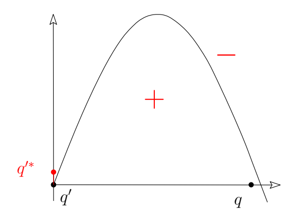
Notice that since $\mathrm{sign}(p((2,0),(x_2,x_3)))$ depends on x_3, the simplified evaluation of Eq. (4) would have given a wrong result:

$$\ell_2 \neq \lim_{\varepsilon_2\to 0^+} \mathrm{sign}(p((2,0),(\varepsilon_2,1))) = \lim_{\varepsilon_2\to 0^+} \mathrm{sign}(\varepsilon_2^2) = 1.$$

The geometric interpretation is that q and q' are both on one of the two lines defined by the quadratic equation $x(y-1)-x^2=0$. Point q' is first slightly moved upwards but this motion leaves it on that same line, then it is moved to the right, and the sign of the quadratic form depends on the vertical position of q' with respect to the other line.

Fourth example: side of a sinusoid

Let $p(\boldsymbol{u}) = (x_1 - \sin x_0)(x_3 - \sin x_2)$ and $\boldsymbol{u}_0 = (q,q') = ((3,0),(0,0))$.
QSP works even if the predicate is non-polynomial. This predicate is positive if q and q' are on the same side of the sinusoid. QSP defines the result for $\mathrm{sign}(p(\boldsymbol{u}_0))$ as



$$\begin{aligned}\mathrm{sign}(p(\boldsymbol{u}_0)) &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+}\lim_{\varepsilon_0\to 0^+} \mathrm{sign}(((\varepsilon_1 - \sin(3+\varepsilon_0))(\varepsilon_3 - \sin\varepsilon_2))\\ &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+}\lim_{\varepsilon_1\to 0^+} \mathrm{sign}((\varepsilon_1 - \sin 3)(\varepsilon_3 - \sin\varepsilon_2))\\ &= \lim_{\varepsilon_3\to 0^+}\lim_{\varepsilon_2\to 0^+} \mathrm{sign}(-(\sin 3)(\varepsilon_3 - \sin\varepsilon_2))\\ &= \lim_{\varepsilon_3\to 0^+} \mathrm{sign}(-(\sin 3)\varepsilon_3) = -1\end{aligned}$$

In this case, the QSP evaluation strategy first computes

$$\ell_1 = \lim_{\varepsilon_3\to 0^+} \mathrm{sign}(p((3,0),(0,\varepsilon_3))) = \mathrm{sign}(-(\sin 3)\varepsilon_3) = -1$$

and it allows us to resolve the degeneracy.

The geometric interpretation is that q'^* is moved slightly above q', so on the side of the sinusoid opposite to q.

2.3 Discussion

Multiple epsilons The idea of utilizing multiple perturbation parameters is already present in Yap's scheme [21], or very recently in Irving and Green's work [14], but without the direct geometric interpretation allowed by QSP. In other previous works, such as SoS [10], the algebraic symbolic perturbation framework was proved to be effective by a careful choice of the exponents for ε, depending on the choice of $G(\boldsymbol{u})$, so as to make some terms negligible. QSP fits in such a traditional framework with a single epsilon, when taking variables ε_ν dependent on a single parameter κ that plays the traditional role of ε. For polynomial predicates, it is enough to take ε_ν exponentially increasing with respect to ν. For example one such choice can be to set $\varepsilon_{N-1} = \kappa$, and $\varepsilon_\nu = \left(\exp\left(\frac{1}{\varepsilon_{\nu+1}}\right)\right)^{-1}$, for $0 \leq \nu < N-1$. The interest of QSP, however, is not to use this view, but rather have the variables ε_ν decoupled; this allows for additional flexibility, and, in particular, permits us to think of the sequence of perturbations in geometric terms.

Efficiency The aim of a perturbation scheme is to solve degeneracies, and a common assumption is that such degeneracies are rare enough so that some extra time can be spent to make a reliable decision when a degeneracy happens. Another implicit assumption is that degeneracies are actually detected, that is, it is implicitly assumed that the original predicates are computed exactly, possibly with some filtering mechanism to ensure efficiency [23].

Nevertheless, the actual additional complexity in case of degeneracy must be addressed. Since QSP is geometrically defined and addresses very general problems, such a complexity analysis cannot be done at the general level. For the two applications described in this paper, the extra predicates needed to resolve the degeneracy have the same complexity as the original ones, while the number of epsilons used to perturb is not bigger than two.

QSP, as many other perturbation schemes, relies on an indexing of the input. However, as mentioned earlier, a given predicate usually depends on a constant number of input objects. It is important to keep in mind that the comparison of indices is necessary only for the few objects involved in a given predicate; sorting the whole input with respect to indices is not required.

Generality In the first three toy examples above, after re-indexing the coordinates $x_i, i = 0, 1, 2, 3$, SoS would have taken $\varepsilon_0 = \varepsilon^8$, $\varepsilon_1 = \varepsilon^4$, $\varepsilon_2 = \varepsilon^2$, and $\varepsilon_3 = \varepsilon$, which yields the same result as QSP. When it leads to a simple result, the classical algebraic view is a very good solution. However, if the original predicate is a bit intricate, the algebraic way will produce numerous extra predicates to resolve degeneracies. Moreover, as for any predicate, some custom work is often still needed on the polynomial to evaluate it efficiently, e.g., finding a good factorization.

QSP provides a very general approach that is able to handle various predicates, even non-polynomial as in the fourth example. Of course applying this scheme to a given problem requires some problem-specific work, but, as noted in the previous paragraph, this is also often the case for the above-mentioned algebraic approaches. In algebraic approaches, obtaining the coefficient of ε^i in a suitable way for an efficient evaluation is a non-trivial task; the task is even harder when the predicate does not boil down to evaluating a single polynomial, as it is the case for Apollonius predicates, which we present in the sequel.

We would not advise the use of QSP for simple cases such as Delaunay triangulations of points where other approaches work well [9], but rather only in cases where the predicates are very complex or non-polynomial. The applications below use high degree polynomials and QSP is a good solution. As far as we know, no other perturbation scheme has ever been proposed for Apollonius diagrams. Regarding intersections of circles, we successfully addressed the predicate comparing the abscissas of intersection of circles, using QSP in Section 3. The only other result that we know of for perturbing this predicate was recently obtained by Irving and Green [14]; it uses a pseudo-random scheme.

Meaningfulness According to a classification by Seidel [20], a perturbation is geometrically meaningful if it allows some control on the direction (in input data space) used to move away from the degeneracies. As QSP combines simple perturbations, it is easier to make it geometrically meaningful as compared to devising an intricate perturbation that must solve a wide range of degeneracy types. For example, for the Apollonius diagram we will design our QSP perturbation to minimize the number of Apollonius vertices (it is also possible to choose to maximize it). QSP is not independent of indexing, but if this indexing is geometrically meaningful, then we can ensure invariance with respect to some geometric transformations.

3 Predicates for circular arcs arrangements

To address the problem of computing arrangements of circular arcs by sweep-line algorithms, two necessary predicates have to compare the abscissae of two endpoints of circular arcs and to decide whether one endpoint of a circular arc lies above or below another circular arc. In this section, we address the x-comparison predicate, which was studied in a previous paper [6]. The other predicate is simpler and we leave as an exercise for the reader to check that the proposed perturbation also resolves its degeneracies.

The arc endpoints are described as intersections of two circles, which leads us to consider the arrangement of all circles supporting arcs or defining their endpoints. Degeneracies occur if several vertices of the arrangement have the same abscissa or if more than two circles meet at a common point. For arrangements exhibiting a lot of degeneracies, it may be interesting to design an algorithm that directly handles special cases, while in other contexts where degeneracies are occasional, it would be preferable to keep the algorithm simple and handle degeneracies through a perturbation scheme.

In the formalism of the previous section, $\boldsymbol{u}$ is a vector of parameters defining a set of circular arcs and $G(\boldsymbol{u})$ is the arrangement. Let us introduce more specific notations for this application. An endpoint $z_{\nu,\mu}$, defined as an intersection of two circles $\mathcal{C}_\nu$ and $\mathcal{C}_\mu$, is determined by the centers (α_ν, β_ν) and (α_μ, β_μ) of the circles (see Figure 1), their squared radii γ_ν and γ_μ, and a Boolean $b_{\nu,\mu}$ encoding whether $z_{\nu,\mu}$ is the leftmost $l_{\nu,\mu}$ or rightmost intersection point $r_{\nu,\mu}$ (if they have the same abscissa, $r_{\nu,\mu}$ is the highest and $l_{\nu,\mu}$ the lowest intersection point).

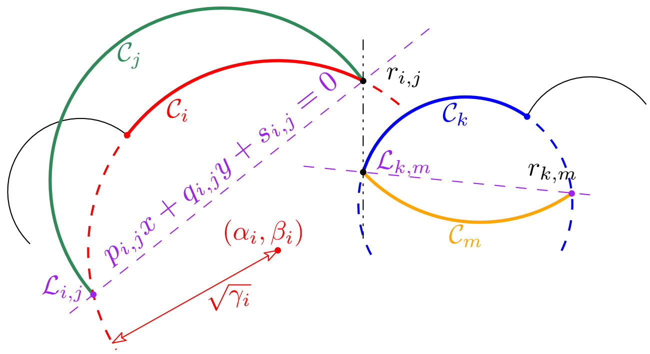


Figure 1: x-comparison of endpoints:
a degenerate case where $r_{i,j}$ and $l_{k,m}$ have the same abcissa.

3.1 Algebraic formulation

Let us introduce the line $\mathcal{L}_{\nu,\mu}$, whose equation $p_{\nu,\mu}x + q_{\nu,\mu}y + s_{\nu,\mu} = 0$ is obtained by subtracting the equations of $\mathcal{C}_\nu$ and $\mathcal{C}_\mu$. We observe that the intersection points of $\mathcal{C}_\nu$ and $\mathcal{C}_\mu$ can also be seen as intersections between $\mathcal{C}_\nu$ and $\mathcal{L}_{\nu,\mu}$. The coefficient of $\mathcal{L}_{\nu,\mu}$ are:

$$p_{\nu,\mu} = 2(\alpha_\nu - \alpha_\mu), \quad q_{\nu,\mu} = 2(\beta_\nu - \beta_\mu), \quad \text{and} \quad s_{\nu,\mu} = \gamma_\nu - \gamma_\mu - \alpha_\nu^2 - \beta_\nu^2 + \alpha_\mu^2 + \beta_\mu^2.$$

The predicate x-$\mathsf{compare}(\mathcal{C}_i, \mathcal{C}_j, b_{i,j}, \mathcal{C}_k, \mathcal{C}_m, b_{k,m})$ compares the abscissae of two arc endpoints $z_{i,j}$ defined by $\mathcal{C}_i$, $\mathcal{C}_j$, and $b_{i,j}$ $(i \neq j)$ on the one hand, and $z_{k,m}$ defined by $\mathcal{C}_k$, $\mathcal{C}_m$, and $b_{k,m}$ $(k \neq m)$ on the other hand. The most complicated evaluation is the sign of the following degree 12 polynomial [6]:

$$(A_{i,j}\, C_{k,m} - A_{k,m}\, C_{i,j})^2 - 4(A_{i,j}\, B_{k,m} - A_{k,m}\, B_{i,j})(B_{i,j}\, C_{k,m} - B_{k,m}\, C_{i,j})$$

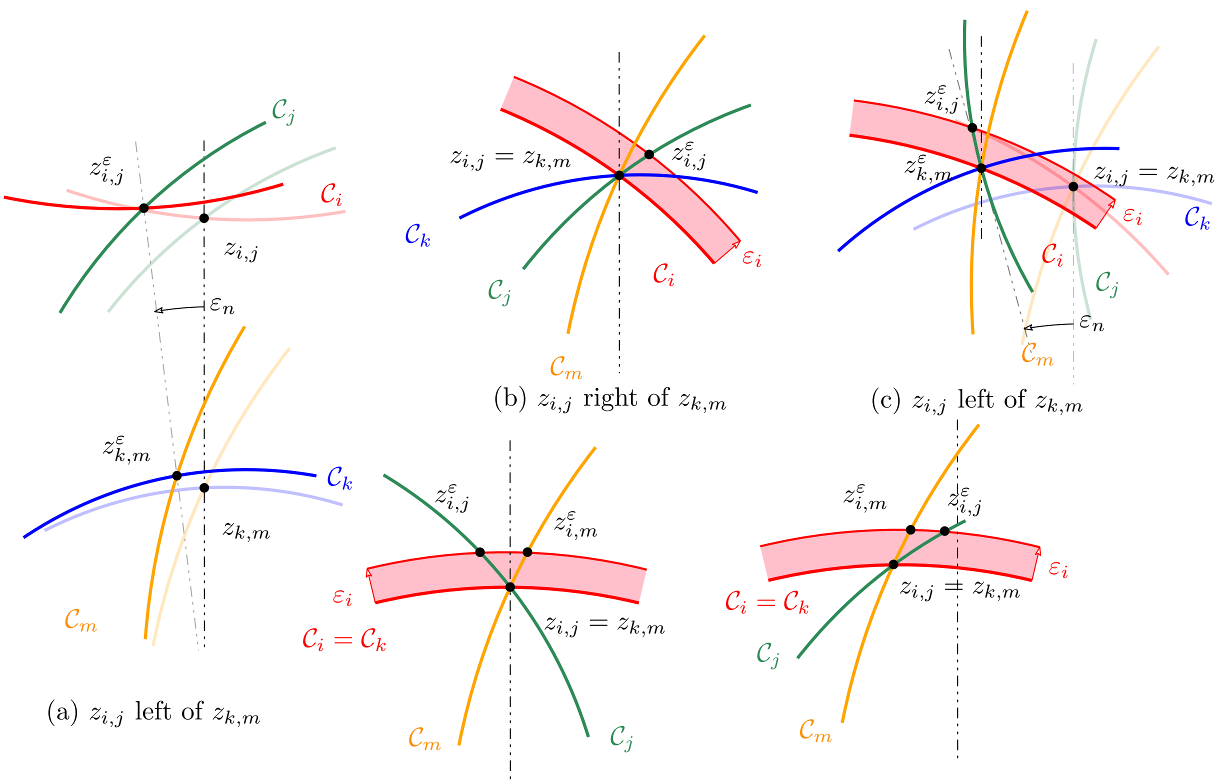
where:

$$\begin{aligned}
A_{\nu,\mu} &= p_{\nu,\mu}^2 + q_{\nu,\mu}^2, \\
B_{\nu,\mu} &= q_{\nu,\mu}^2\, \alpha_\nu - p_{\nu,\mu}(\, s_{\nu,\mu} + q_{\nu,\mu}\, \beta_\nu), \\
C_{\nu,\mu} &= (s_{\nu,\mu} + q_{\nu,\mu}\, \beta_\nu)^2 + q_{\nu,\mu}^2\, (\alpha_\nu^2 - \gamma_\nu).
\end{aligned}$$

Introducing an algebraic symbolic perturbation yields quite a complicated polynomial in ε, not really suitable for efficient predicate evaluation. For the same problem, Irving and Green [14] use an algebraic perturbation with pseudo-random coefficients, but they only address the special case where $\mathcal{C}_i = \mathcal{C}_k$.

3.2 Qualitative symbolic perturbation

We construct a sequence of $N = n + 1$ successive perturbations for an input $\boldsymbol{\mathcal{C}}$ consisting of n circles $\mathcal{C}_\nu$, $0 \leq \nu < n$.



(a) $z_{i,j}$ left of $z_{k,m}$

(b) $z_{i,j}$ right of $z_{k,m}$

(c) $z_{i,j}$ left of $z_{k,m}$

(d) $z_{i,j}$ left of $z_{i,m}$

(e) $z_{i,j}$ right of $z_{i,m}$

Figure 2: Perturbing $z_{i,j}$ and $z_{k,m}$

The first perturbation, $\pi_n(\boldsymbol{\mathcal{C}}, \varepsilon_n)$ is a rotation centered at the origin and with angle ε_n. This perturbation handles the cases where $z_{i,j}$ and $z_{k,m}$ are different points with the same abscissa. Actually if $z_{i,j} \neq z_{k,m}$, with the notations of Theorem 1, $\ell_1 \neq 0$ and the sign of the perturbed predicate is the one of ℓ_1. In other words, it just uses lexicographic comparisons: y-comparisons are used to break ties in x-comparisons (see Figure 2(a)). The rotation by a small angle has the same effect as a shear transform. A shear transform is often preferred in the literature because it is a rational transformation. We use a rotation instead because it transforms circles into circles, which is preferable in order to apply the remaining perturbations in the sequence. On top of that, we are not interested in the algebraic formulation of the transformation, since we look at the limit in a geometric way instead of algebraically.

In fact, the next perturbations alone may remove the degenerate x-comparisons, but the main interest of adding this first perturbation is to treat these degeneracies in a meaningful way.

We are now left with cases when $z_{i,j} = z_{k,m}$. The other perturbations in the sequence consist in inflating the circles: $\pi_\nu(\boldsymbol{\mathcal{C}}, \varepsilon_\nu)$ replaces γ_ν by $\gamma^{\varepsilon}_\nu = \gamma_\nu + \varepsilon_\nu$, $\varepsilon_\nu \geq 0$. Recall that the comparison of indices is necessary only for the four circular arcs involved in a given predicate; sorting the whole input with respect to indices is not required. We consider the circles by decreasing radii to get rid of pairs of tangent circles in a geometrically meaningful way: if two circles are tangent, the perturbation inflates the largest one by a larger amount, making the intersection point either disappear if the two circles are internally tangent, or split into two points if they are externally tangent.

Without loss of generality, we may assume that $i \geq j, k, m$, i.e., $\mathcal{C}_i$ is the most perturbed circle. If $i \notin \{k, m\}$, then $z_{i,j}$ moves (i.e., $z^{\varepsilon}_{i,j} \neq z_{i,j}$) while $z_{k,m}$ stays fixed. If $\mathcal{C}_i$ and $\mathcal{C}_j$ have a non-vertical tangent in $z_{i,j}$ and after determining the vertical order of $\mathcal{C}_i$ and $\mathcal{C}_j$ at the right of $z_{i,j}$, and whether $z_{i,j}$ is on the top or bottom part of $\mathcal{C}_i$ using the auxiliary predicates described below, it is easy to decide if $z^{\varepsilon}_{i,j}$ moves left or right when $\varepsilon_i > 0$ (see Figure 2(b)). With the notations of Theorem 1, $\ell_1 = 0$ since $z_{i,j} = z_{k,m}$, $\ell_2, \ldots, \ell_{N-i-1} = 0$ since the corresponding perturbations do not affect a circle involved in the current predicate and ℓ_{N-i} is the first non-zero value in the (ℓ_μ) sequence. Furthermore, the side $z^{\varepsilon}_{i,j}$ is not dependent from the initial rotation nor the unperturbed circles, thus $\ell_{N-i} = \ell'_{N-i}$.

If $\mathcal{C}_i$ or $\mathcal{C}_j$ have a vertical tangent in $z_{i,j}$, then we have to consider the rotation to find the sign of the perturbed slope before determining the side of $z^{\varepsilon}_{i,j}$ (see Figure 2(c)). With respect to Theorem 1, ℓ_{N-i} is still the first non-zero value in the sequence (ℓ_μ) but ℓ_{N-i} can be different from ℓ'_{N-i} .

If $i \in \{k, m\}$, assume, without loss of generality, that $i = k$. If $z^{\varepsilon}_{i,j}$ and $z^{\varepsilon}_{i,m}$ are perturbed in opposite x-directions, it is easy to decide which one is the leftmost (see Figure 2(d)). Otherwise, we determine the vertical order of the three circles $\mathcal{C}_i$, $\mathcal{C}_j$, and $\mathcal{C}_m$ at the right of $z_{i,j} = z_{i,m}$; we know that $\mathcal{C}_i$ is either the topmost or the bottommost circle in this vertical ordering. The point lying on the closest arc to $\mathcal{C}_i$ is more perturbed than the other, and the auxiliary predicates below allow us to decide which of $z^{\varepsilon}_{i,j}$ and $z^{\varepsilon}_{i,m}$ is to the left (see Figure 2(e)). As in the previous case, Theorem 1 applies and ℓ_{N-i} is the first non-zero value, and $\ell_{N-i} = \ell'_{N-i}$ if there is no vertical tangent in $z_{i,j}$.

Auxiliary predicates A circle can be split in four parts: top-right, top-left, bottom-left, and bottom-right at its points with horizontal or vertical tangents. Knowing if a point $z_{i,j}$ is on the left or right part of $\mathcal{C}_i$ can be evaluated by x-compare$(\mathcal{C}_i, \mathcal{C}_j, b_{i,j}, \mathcal{C}_i, \mathcal{C}_m, b_{i,m})$ for a suitable $\mathcal{C}_m$ such that $\mathcal{L}_{i,m}$ has equation $x - \alpha_i = 0$. Discriminating between the top and bottom parts is done in the same way, by exchanging the roles of the x- and y-coordinates. Another predicate consists in deciding if $\mathcal{C}_i$ is above or below $\mathcal{C}_j$ at the right of $r_{i,j}$; this can be done by elementary geometric computations.

4 The Apollonius diagram

In this section, we describe how QSP applies to predicates related to the Apollonius diagram in dimensions 2 and 3.

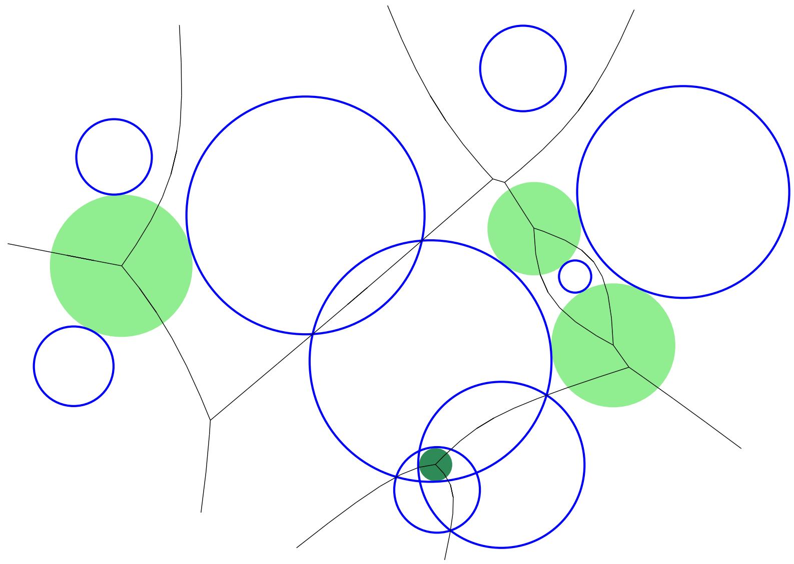


Figure 3: Planar Apollonius diagram. Weighted points are in light blue. A green disk is centered at an Apollonius vertex and its radius is the weighted distance of the center to its three closest sites. The darker green disk has a negative distance to its closest neighbors.

4.1 Definition

The *Apollonius diagram*, also known as *additively weighted Voronoi diagram*, is defined on a set of weighted points in the Euclidean space $\mathbb{R}^d$. In the formalism of Section 2, $\boldsymbol{u}$ is a vector of coordinates and weights of a set of weighted points and $G(\boldsymbol{u})$ is the Apollonius diagram; as in Section 3, we introduce additional specific notations. The Euclidean norm is denoted as $|\cdot|$. The *weighted distance* from a query point q to a weighted point (p, w), where p is a point in the Euclidean space and $w \in \mathbb{R}$, is $|pq| - w$. The Apollonius diagram is the closest point diagram for this distance. It generalizes the Voronoi diagram, defined on non-weighted points.

Given a set of weighted points, also called *sites*, adding the same constant to all weights does not change the Apollonius diagram. Thus, in the sequel, we may freely translate the weights to ensure, for example, that all weights are positive, or that a particular weight is zero. A site $(s, w), w \geq 0$, can be identified with the sphere S centered at s and of radius w. The distance from a query point r to a site $S = (s, w)$ is the Euclidean distance from r to S, with a negative sign if r lies inside S.

An Apollonius vertex v is a point at the same distance from $d+1$ sites $S_0, S_1, \ldots, S_d$ in general position. We call the configuration *external* if v is outside sphere S_i, for all $i = 0, \ldots, d$, and *internal* if it is inside the spheres. If the configuration is external (resp., internal), v is the center of a sphere externally (resp., internally) tangent to the sites S_i (see green (resp., dark green) disks in Figure 3). It is always possible to ensure an external configuration locally by adding a suitable constant to the weights of all S_i, such that all weights are non-negative, while the smallest among them is equal to zero.

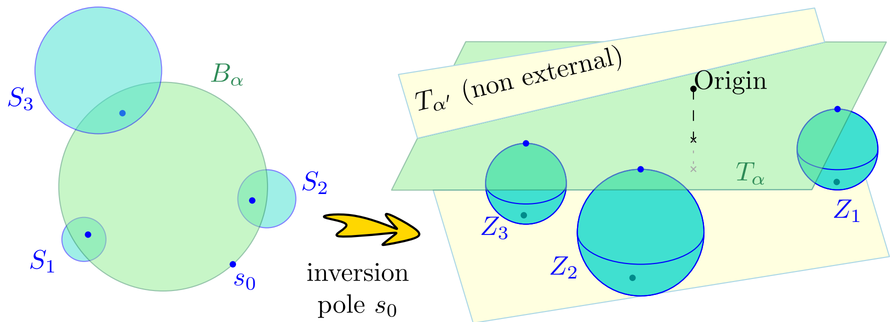


Figure 4: The spheres B_α externally tangent to the sites S_i, $i = 0, \ldots, d$, correspond, via the inversion transformation with s_0 as the pole, to hyperplanes T_α tangent to the spheres Z_i that separate them from the origin.

Let us show that $d+1$ sites in general position define 0, 1 or 2 Apollonius vertices. Assume, without loss of generality, that all weights are non-negative for $i = 1, \ldots, d$ and $w_0 = 0$, so as to have an external configuration. Consider now the inversion with point s_0 as the pole. The point s_0 goes to infinity, while each sphere S_i, $i = 1, \ldots, d$ becomes a new sphere $Z_i = (z_i, \rho_i)$ (see Figure 4). Determining the balls B_α (where α indexes the different solutions) tangent to the spheres S_i, $i = 0, \ldots, d$ is equivalent to determining halfspaces delimited by the hyperplanes T_α tangent to the spheres Z_i, $i = 1, \ldots, d$, with all spheres on the same side of T_α. Requiring that a given B_α is externally tangent to the spheres S_i is equivalent to requiring that T_α separates the spheres Z_i from the origin. The normalized equation of T_α: $\lambda_\alpha \cdot x + \delta_\alpha = 0$, with $\lambda_\alpha \in \mathbb{R}^d$,$|\lambda_\alpha| = 1$ and $\delta_\alpha \in \mathbb{R}$, gives the signed distance of a point $x \in \mathbb{R}^d$ to T_α. We have

$$T_\alpha \text{ tangent to } Z_i,\ 1 \leq i \leq d \quad \Longleftrightarrow \quad \left\{ \begin{array}{l} \lambda_\alpha \cdot z_i + \delta_\alpha = \rho_i,\ 1 \leq i \leq d \\ |\lambda_\alpha|^2 = 1 \end{array} \right. . \tag{5}$$

In the inverted space, the general position hypothesis means that the spheres Z_i do not have an infinity of tangent hyperplanes; the latter can occur only if the points z_i (and thus the points s_i) are affinely dependent. Therefore, the system (5) of one quadratic and d linear equations in $d+1$ unknowns ($\lambda_\alpha \in \mathbb{R}^d$ and $\delta_\alpha \in \mathbb{R}$) has at most two real solutions by Bézout's theorem, hence the first claim follows. Depending on the position of the origin with respect to T_α (or equivalently on the sign of δ_α), zero, one or both solutions may correspond to external configurations.

An Apollonius vertex is actually defined by a sequence of $d+1$ sites in general position, up to a positive permutation of the sequence. Indeed, in the previous paragraph, if there are two solutions T_α and $T_{\alpha'}$, we observe that they are symmetric with respect to the hyperplane spanned by the points z_i, thus the d-simplex formed by the tangency points and the origin has different orientations for the two solutions (see Figure 5). This implies that the two solutions can be distinguished by the signature of the permutation of the spheres S_i.

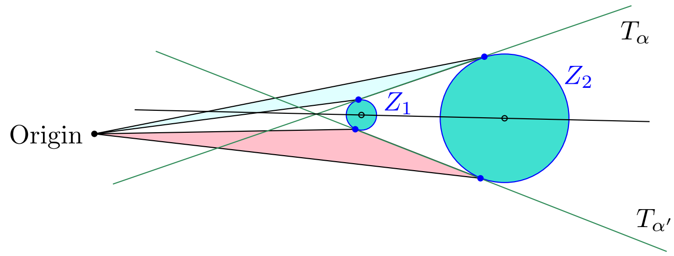


Figure 5: If T_α and $T_{\alpha'}$ are both external, the simplices formed by the tangency points and the origin have different orientations.

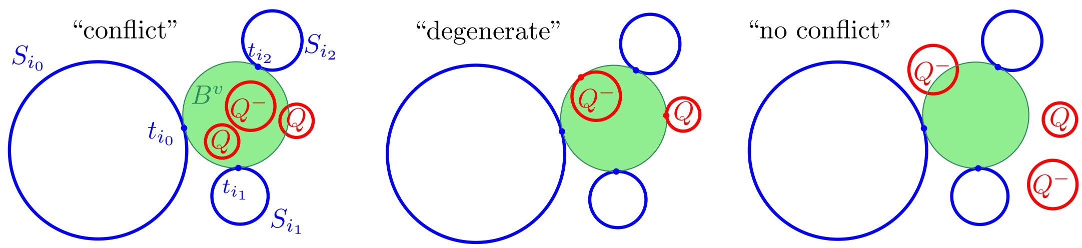


Figure 6: Examples of positions of Q or Q^- that are in "conflict", "degenerate" or "no conflict" configuration with an Apollonius vertex v.

4.2 The VConflict predicate

We start with the *vertex conflict* predicate $\mathsf{VConflict}(\boldsymbol{S}^v, Q)$, which answers the following question:

> *Does an Apollonius vertex v defined, up to a positive permutation, by a $(d+1)$-tuple of sites $\boldsymbol{S}^v = (S_{i_0}, S_{i_1}, \dots, S_{i_d})$ remain as a vertex of the diagram after another site Q is added?*

This predicate is intrinsic to computing an Apollonius diagram: it boils down to computing the diagram of the $d+2$ sites, thus any algorithm able to compute the diagram of these sites allows to answer the predicate. If the site centered at v and tangent to the sites of the tuple $\boldsymbol{S}^v$ is in internal configuration, we can add a negative constant to the radii of all spheres in $\boldsymbol{S}^v \cup \{Q\}$ so that the smallest site in $\boldsymbol{S}^v$ has zero radius. Then the configuration of the common tangent sphere becomes external. In this manner, we can always restrict our analysis to the case where the Apollonius vertex we consider is in external configuration. Note that this may lead to a negative weight w_q for Q, which was *a priori* excluded above, but is treated below.

We denote by $B_{i_0 i_1 \dots i_d}$ the open ball whose closure $\overline{B}_{i_0 i_1 \dots i_d}$ is tangent to the sites of $\boldsymbol{S}^v$; its boundary is denoted $\partial B_{i_0 i_1 \dots i_d}$. The contact points $t_{i_0}, t_{i_1}, \dots, t_{i_d}$ define a positively oriented d-simplex.

If $w_q \geq 0$, the predicate $\mathsf{VConflict}(\boldsymbol{S}^v, Q)$ answers (Figure 6)

- "conflict" if Q intersects $B_{i_0 i_1 \dots i_d}$,
- "no conflict" if Q and $\overline{B}_{i_0 i_1 \dots i_d}$ are disjoint,
- "degenerate" if Q and $B_{i_0 i_1 \dots i_d}$ do not intersect, while Q and $\overline{B}_{i_0 i_1 \dots i_d}$ are tangent.

If $w_q < 0$, we define Q^- as the sphere with the same center s_q as Q and radius $-w_q$. Then $\mathsf{VConflict}(\boldsymbol{S}^v, Q)$ answers

- "conflict" if Q^- is included in $B_{i_0 i_1 \dots i_d}$,
- "no conflict" if Q^- intersects the complement of $\overline{B}_{i_0 i_1 \dots i_d}$,
- "degenerate" if Q^- is included in $\overline{B}_{i_0 i_1 \dots i_d}$ and is tangent to its boundary.

Qualitative perturbation of the VConflict predicate QSP relies on some ordering of the sites. Each site $S_\nu = (s_\nu, w_\nu)$ is perturbed to $S_\nu^\varepsilon = (s_\nu, w_\nu + \varepsilon_\nu)$, $\varepsilon_\nu \geq 0$, with S_λ perturbed more than S_ν if $\lambda > \nu$. Following the QSP framework, if the configuration is still degenerate after we have enlarged the site of maximum index, then we enlarge the site with the second largest index, and so on. As mentioned in the general presentation (Section 2.3), we need only consider the sites involved in the predicate, and enlarge them one-by-one until the resulting configuration is non-degenerate, in which case the predicate is resolved. Sites are sorted internally in the predicate, among a constant number of objects; there is no need for any global sorting of the sites.

Any indexing can be used. We choose what we call the *max-weight* indexing, which assigns a larger index to the site with larger weight. As a result, a site with larger weight is perturbed more, and in order to resolve the predicate we need to consider the sites in order of decreasing weights, until the degeneracy is resolved. To break ties between sites with the same weights, we use the lexicographic comparison of their centers: among two sites with the same weight, the site whose center is lexicographically smaller than the other is assigned a smaller max-weight index. The max-weight indexing has the strong advantage of being geometrically meaningful. It favors sites with larger weights, so, if two sites are internally tangent, then the site with the larger weight will be perturbed more, in which case the site with the smallest weight will be inside the interior of the other site, and its Apollonius region will disappear in the perturbed diagram. As a first consequence, this indexing minimizes the number of Apollonius regions in the diagram, or, equivalently it maximizes the number of hidden sites in the diagram. Secondly, and most importantly, the tangency points of the sites with the Apollonius circles that they define in the diagram are pairwise distinct. This property makes the analysis of the perturbed predicates much simpler, whereas the Apollonius diagram computed does not exhibit pathological cases, such as Apollonius regions with empty interiors. Some inevitable degenerate constructions, such as zero-length Apollonius edges, are handled seamlessly by the method. As a final comment, the max-weight scheme can be used to resolve the degeneracies of all predicates described by Emiris and Karavelas for the 2D case [12].

4.3 Perturbing circles for the 2D Apollonius diagram

In two dimensions, the two main predicates for computing Apollonius diagrams are the VConflict predicate introduced in the previous section and the EConflict predicate, which will be analyzed in Section 4.3.5. Other predicates deal with special cases such as infinite Apollonius edges and can be treated in a similar manner.

4.3.1 Algebraic expression for the 2D VConflict predicate

Let S_i, S_j, S_k be the three sites that define an Apollonius circle in the Apollonius diagram and let $Q = S_q$ be the query site.

Let us first define some intermediate quantities:

$$x^*_\nu = x_\nu - x_i, \quad y^*_\nu = y_\nu - y_i, \quad w^*_\nu = w_\nu - w_i, \quad p^*_\nu = (x^*_\nu)^2 + (y^*_\nu)^2 - (w^*_\nu)^2, \quad \nu \in \{j,k,q\},$$

$$E_s = \begin{vmatrix} s^*_j & p^*_j \\ s^*_k & p^*_k \end{vmatrix}, \qquad E_{st} = \begin{vmatrix} s^*_j & t^*_j & p^*_j \\ s^*_k & t^*_k & p^*_k \\ s^*_q & t^*_q & p^*_q \end{vmatrix}, \qquad s,t \in \{x,y,w\},$$

and

$$\Delta = (E_x)^2 + (E_y)^2 - (E_w)^2.$$

In the algebraic formulation of the predicate by Emiris and Karavelas [12], the evaluation of VConflict(S_i, S_j, S_k, Q) relies on the computation of the sign of the expression I:

$$I := E_{xw}E_x + E_{yw}E_y + E_{xy}\sqrt{\Delta}.$$

The expression I is of the form $X_0 + X_1\sqrt{Y}$, where the algebraic degrees of X_0, X_1 and Y are 7, 4 and 6, respectively. Its sign can be computed by means of the formula

$$\operatorname{sign}(X_0 + X_1\sqrt{Y}) = \begin{cases} \operatorname{sign}(X_1) & \text{if } X_0 = 0 \\ \operatorname{sign}(X_0) & \text{if } X_1 = 0 \text{ or } Y = 0 \\ \operatorname{sign}(X_0) & \text{if } \operatorname{sign}(X_0) = \operatorname{sign}(X_1) \\ \operatorname{sign}(X_0)\operatorname{sign}(X_0^2 - X_1^2 Y) & \text{otherwise.} \end{cases} \tag{6}$$

It is thus concluded by Emiris and Karavelas that the algebraic degree of the predicate is 14 [12, Theorem 11].

In fact we can further decrease the algebraic degree of the predicate by observing that the quantity $X_0^2 - X_1^2 Y$ can be factorized as follows (see [8, Appendix B] for the details of this derivation):

$$X_0^2 - X_1^2 Y = [(E_x)^2 + (E_y)^2]\,[(E_{xw})^2 + (E_{yw})^2 - (E_{xy})^2],$$

where the first factor is a non-negative quantity of degree 6, whereas the second factor is of degree 8. In fact, when we compute the sign of the quantity $X_0^2 - X_1^2 Y$, we already know that the quantity $(E_x)^2 + (E_y)^2$ is strictly positive, since otherwise X_0 would have been zero (X_0 is a linear combination of E_x and E_y), which has already been ruled out according to the procedure in Eq. (6). Hence the algebraic degree of the predicate is 8.

Before using our qualitative symbolic perturbation framework to design the perturbed predicate, we briefly sketch how a standard algebraic perturbation framework could be applied.

4.3.2 Algebraic perturbation of the 2D VConflict predicate

If $S_\nu = (x_\nu, y_\nu, w_\nu)$ is perturbed in $S_\nu^\varepsilon = (x_\nu, y_\nu, w_\nu + \varepsilon_\nu)$ for $\nu \in \{i, j, k, q\}$, then developing an expression like $(E_{xw})^2$ will give a polynomial of degree 6 in ε_i, ε_j, ε_k and ε_q with 186 terms. Assigning ε_i, ε_j, ε_k and ε_q to be polynomial functions of a single variable ε (for example, we may set $\varepsilon_\nu = \varepsilon^{\alpha_\nu}$, $\nu \in \{i, j, k, q\}$) transforms the expression to a univariate polynomial in ε. When performing such an assignment, either some of the terms collapse making their geometric and algebraic interpretation difficult, or α_i, α_j, α_k and α_q have to be chosen carefully so that the coefficients of the various monomials of the variables ε_ν in the resulting polynomial do not collapse. Even if one could find an assignment that does not make the coefficients (of the originally different terms) collapse, we are still faced with the problem of analyzing the monomials, and, by employing algebraic and/or geometric arguments, showing that there is at least one coefficient of the polynomial that does not vanish.

4.3.3 Qualitative perturbation of the 2D VConflict predicate

We now precisely describe how the perturbation works on the VConflict predicate in dimension 2. What follows in this section does not depend on the actual algebraic formulation of the VConflict predicate; instead our entire analysis is based on purely geometric arguments.

Let us denote by q the max-weight index of Q, i.e., $Q = S_q$. We denote with superscript ε the perturbed version of objects, that is B_{ijk}^ε is a shorthand for the ball tangent to S_i^ε, S_j^ε, and S_k^ε, and

$$\mathsf{VConflict}^\varepsilon(S_i, S_j, S_k, S_q) = \lim_{\varepsilon_{i_3}\to 0^+} \lim_{\varepsilon_{i_2}\to 0^+} \lim_{\varepsilon_{i_1}\to 0^+} \lim_{\varepsilon_{i_0}\to 0^+} \mathsf{VConflict}(S_i^\varepsilon, S_j^\varepsilon, S_k^\varepsilon, Q^\varepsilon)$$

with $i_0 < i_1 < i_2 < i_3$, $\{i_0, i_1, i_2, i_3\} = \{i, j, k, q\}$.

If $q > i, j, k$ and VConflict $(S_i, S_j, S_k, Q) =$ "degenerate", we compute the limit given by Eq. (2)

$$\ell_1 = \lim_{\varepsilon_q \to 0^+} \mathsf{VConflict}(S_i, S_j, S_k, Q^\varepsilon).$$

It is clear that this limit always evaluates to "conflict", since Q is growing while the open ball B_{ijk} whose closure is tangent to S_i, S_j, and S_k can be considered as fixed. Thus we can apply Theorem 1 and we do not need to look at perturbations of smaller indices (see Figure 7).

If q is not the largest index, then B_{ijk} can be viewed as defined by three other circles among S_i, S_j, S_k, and Q. Since $B_{ijk} = B_{jki} = B_{kij}$, we can assume, without loss of generality, that $i > j, k, q$. Moreover, B_{ijk} coincides with either B_{jkq} or B_{kjq}, depending on the orientation of the tangency points of S_j, S_k and Q with B_{ijk}.

In the perturbed setting, S_i^ε is in conflict with B_{jkq}^ε (or B_{kjq}^ε) since S_i^ε is growing, while B_{jkq}^ε can be considered as fixed. We simply need to determine if B_{ijk}^ε remains empty in the perturbed setting. Let t_ν (resp., t_q) be the tangency point of S_ν (resp., Q) with B_{ijk},

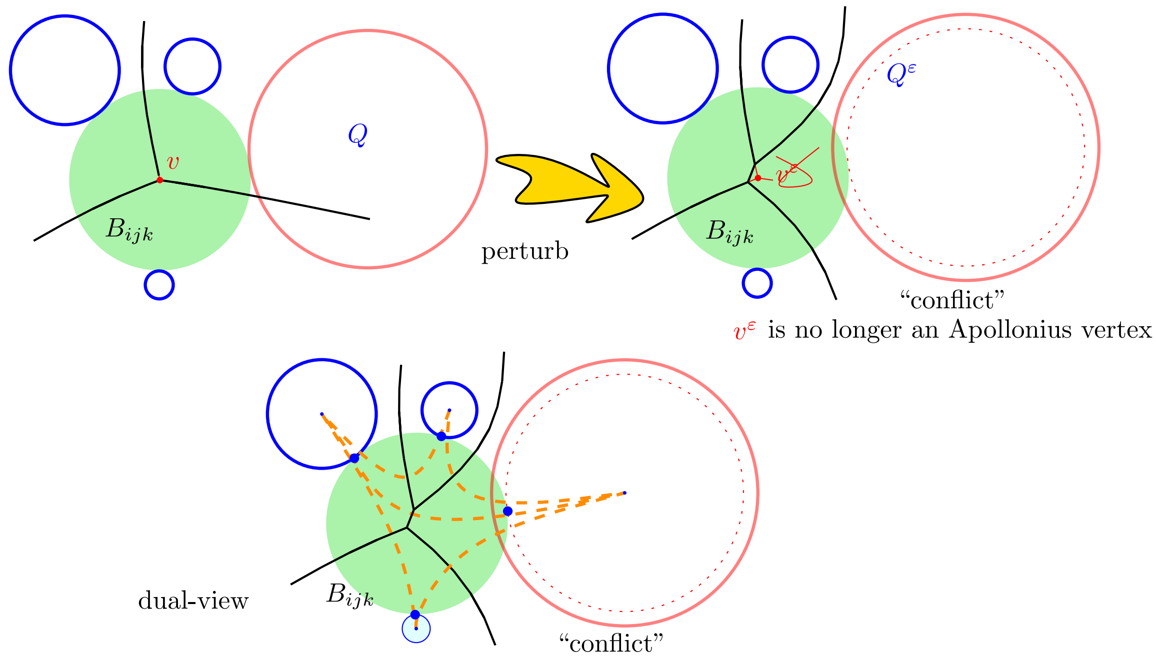


Figure 7: Perturbing a degenerate Apollonius vertex: the case $q > i, j, k$.

$\nu \in \{i, j, k\}$, and notice that $t_i t_j t_k$ is a ccw triangle. We consider three cases depending on the position of t_q on ∂B_{ijk}. If t_q is different from t_i, t_j, and t_k, the four points form a convex quadrilateral. When perturbing S_i to become S_i^{ε}, the Apollonius vertex is split in two, which, in the dual,[1] corresponds to a triangulation of the quadrilateral with vertices S_i, S_j, S_k, S_q. Since S_i is the most perturbed circle, the quadrilateral will be triangulated by linking S_i to the other three vertices. If t_q is on the same side as t_i with respect to the line $t_j t_k$, then the triangulation contains triangle $S_i S_j S_k$ and, therefore, Q is not in conflict with B_{ijk} (see Figure 8), otherwise $S_i S_j S_k$ is not in the triangulation and Q has to be in conflict with B_{ijk} (see Figure 9).

If t_q is equal to t_i then, since $i > q$, the site Q is internally tangent to S_i and there is no conflict (S_i^{ε} contains Q in its interior, and thus Q has empty Apollonius region in the diagram). If t_q is equal to t_ν with $\nu \in \{j, k\}$ then either Q is internally tangent to S_ν, or S_ν is internally tangent to Q. In the former case, Q does not intersect the perturbed Apollonius disk B_{ijk}^{ε} and thus the result of the perturbed predicate is "no conflict"; in the latter case, Q intersects B_{ijk}^{ε}, and the perturbed predicate returns "conflict". Hence, in the case $t_q = t_\nu$, $\nu \in \{j, k\}$, the perturbed predicate returns "conflict" if an only if $q > \nu$.

[1] The dual of the Apollonius diagram is called Apollonius graph. The Apollonius region of a site S_i is associated to a vertex of the dual graph, thus S_i can be used to refer to the corresponding vertex in the dual graph.

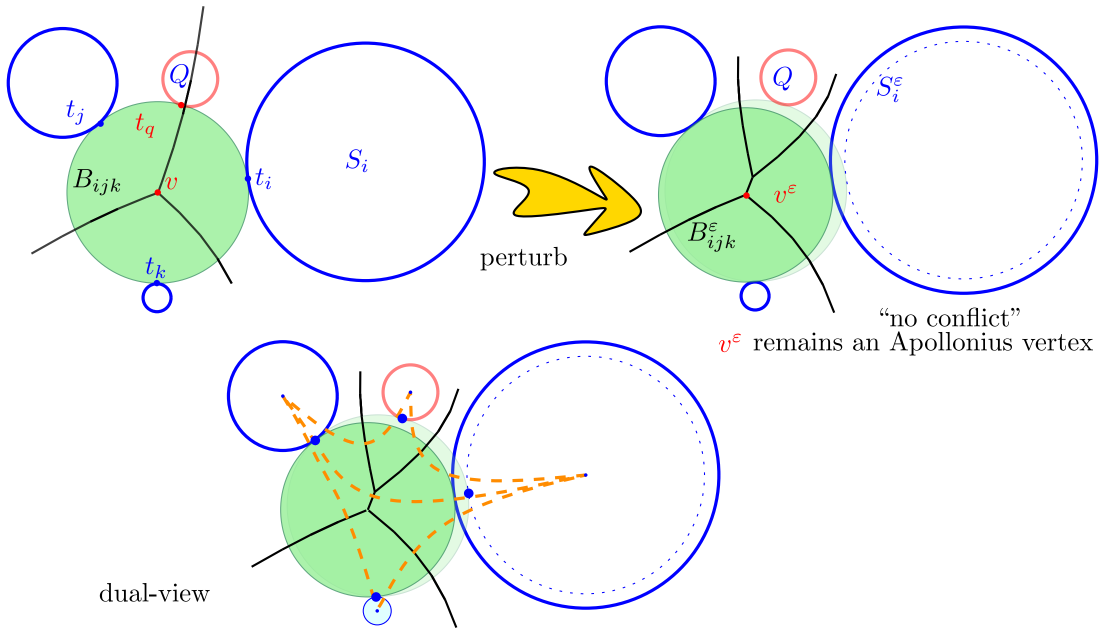


Figure 8: Perturbing a degenerate Apollonius vertex: the case $i > j, k, q$ and $t_j t_k t_q$ is a ccw triangle.

4.3.4 Practical evaluation of the 2D $\mathsf{VConflict}^\varepsilon$ predicate

Following the analysis in the previous section, $\mathsf{VConflict}^\varepsilon(S_i, S_j, S_k, Q)$ can be evaluated by the following procedure:

1. **if** $\mathsf{VConflict}(S_i, S_j, S_k, Q) \neq$ "degenerate" **then return** $\mathsf{VConflict}(S_i, S_j, S_k, Q)$;
2. **if** $q > \max\{i, j, k\}$ **then return** "conflict";
3. ensure that $i > \max\{j, k\}$ by a cyclic permutation of (i, j, k);
4. **if** $t_q = t_i$ **then return** "no conflict";
5. **if** $t_q = t_j$ **then** { **if** $q > j$ **then return** "conflict"; **else return** "no conflict"; };
6. **if** $t_q = t_k$ **then** { **if** $q > k$ **then return** "conflict"; **else return** "no conflict"; };
7. **if** $t_j t_k t_q$ is ccw **then return** "no conflict"; **else return** "conflict";

Step 1 is evaluated as described in Section 4.3.1. Steps 2 and 3 amount to sorting the indices (i.e., the weights) of the four sites and determining if q is the largest, or, if this is not the case, finding the largest index. At Step 4, we already know that $i > q$, which implies that $w_i \geq w_q$, and hence the only possibility is that Q is internally tangent to S_i. So, in order to perform Step 4, we simply look at $p_q^* = (x_q - x_i)^2 + (y_q - y_i)^2 - (w_q - w_i)^2$: if $p_q^* = 0$, return "no conflict", otherwise continue with Step 5. Steps 5 and 6 can be resolved in a similar way: if $(x_q - x_\nu)^2 + (y_q - y_\nu)^2 - (w_q - w_\nu)^2 = 0$, then if $q > \nu$ (resp., $q < \nu$), we return "conflict" (resp., "no conflict"). Otherwise, we continue with the last step of the procedure.

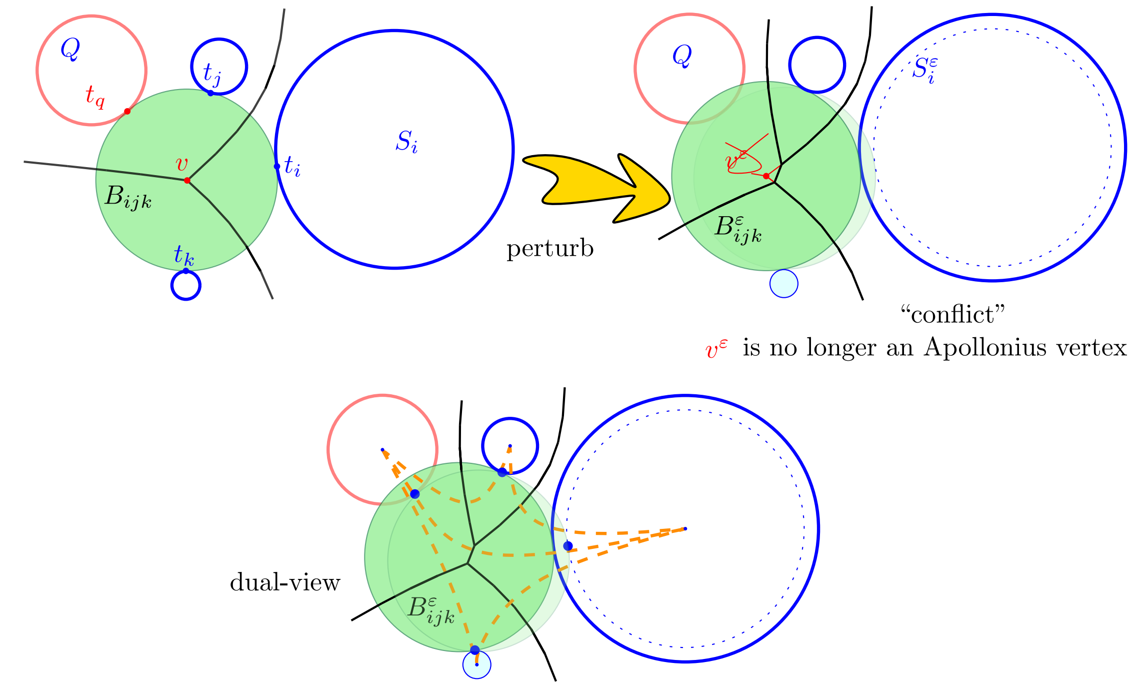


Figure 9: Perturbing a degenerate Apollonius vertex:
the case $i > j, k, q$ and $t_j t_k t_q$ is a cw triangle.

We will now focus on this last step, Step 7, because it introduces a new geometric predicate, which is difficult to evaluate: $\mathsf{Orientation}(t_j, t_k, t_q)$, for three tangency points. Our aim is to reduce the complexity of the expressions to be evaluated, which is why we avoid computing the tangency points explicitly. The end of this section describes a method with algebraic degree 8, as in Step 1. This computation can be done in another way: in [8, Appendix A] we proposed an alternative method that requires very few additional arithmetic computations besides the quantities already computed in the unperturbed evaluation of Step 1, however these few extra computations have algebraic degree 12.

It has been shown in [12] that evaluating the orientation of three points where two are centers of sites and the third is an Apollonius vertex, can be performed using algebraic expressions of degree at most 14. In fact, this degree may be decreased to 8 (see [8, Appendix B]), in which case we resolve $\mathsf{Orientation}(t_j, t_k, t_q)$, without resorting to a higher degree predicate, as described below.

Firstly, we evaluate $o_1 = \mathsf{Orientation}(s_j, v_{ijk}, s_k)$, where v_{ijk} is the center of the Apollonius circles B_{ijk} of the three sites S_i, S_j, S_k. We perform this evaluation in order to determine whether the angle α of the ccw arc $\widehat{t_j t_k}$ on B_{ijk} is more or less than π. Secondly, we distinguish between the following cases (see Figure 10):

$o_1 =$ "collinear". In this case $\alpha = \pi$, and the line through t_j and t_k coincides with the line through s_j and s_k. Hence: $\mathsf{Orientation}(t_j, t_k, t_q) = \mathsf{Orientation}(s_j, s_k, s_q)$ (see Q_n

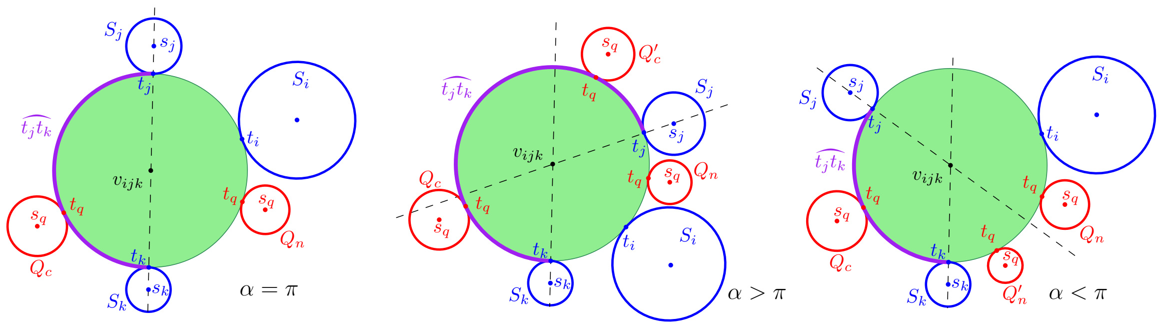


Figure 10: Computing the auxiliary predicate Orientation(t_j, t_k, t_q) using orientations involving the Apollonius vertex v_{ijk}. From left to right: the case $\alpha = \pi$, the case $\alpha > \pi$, and the case $\alpha < \pi$, where α is the angle of the ccw oriented arc $\widehat{t_jt_k}$.

(resp., Q_c) in Figure 10(left) to illustrate a position of Q not in conflict (resp., in conflict)).

$o_1 =$ "ccw". In this case $\alpha > \pi$. We start by evaluating $o_2 =$ Orientation(s_j, v_{ijk}, s_q). If $o_2 \neq$ "ccw" (see Q'_c in Figure 10(middle)), t_q lies to the right of the line through t_j and t_k, and thus Orientation$(t_j, t_k, t_q) =$ "cw". Otherwise, we need to evaluate the orientation $o_3 =$ Orientation(v_{ijk}, s_k, s_q); then Orientation$(t_j, t_k, t_q) =$ "ccw" if and only if $o_3 =$ "ccw" (see Q_n and Q_c in Figure 10(middle)).

$o_1 =$ "cw". In this case $\alpha < \pi$. We start by evaluating $o_2 =$ Orientation(s_j, v_{ijk}, s_q). If $o_2 \neq$ "cw", t_q lies to the left of the line through t_j and t_k, and thus Orientation$(t_j, t_k, t_q) =$ "ccw" (see Q_n in Figure 10(right)). Otherwise, we need to evaluate the orientation $o_3 =$ Orientation(v_{ijk}, s_k, s_q); then Orientation$(t_j, t_k, t_q) =$ "cw" if and only if $o_3 =$ "cw" (see Q_c and Q'_n in Figure 10(right)).

To summarize, the evaluation of Step 7 requires at most three orientation tests involving an Apollonius vertex and two sites; one may be obtained as a subproduct of Step 1, while the other two require work similar to the work performed for Step 1. Thus, the evaluation of Step 7 does not increase the algebraic degree of the VConflict predicate.

4.3.5 Qualitative perturbation for the 2D EConflict predicate

The computation of the 2D Apollonius diagram requires another predicate. When a new site is added it is not enough to find which Apollonius vertices remain or disappear: we need a more complete analysis of the modification of the edges of the diagram. This is the subject of the EConflict predicate. Given four sites S_i, S_j, S_k and S_l that define a Voronoi edge e in the diagram, and a query site Q, EConflict determines the type of conflict of Q with the edge e. This predicate boils down to computing the diagram of the 5 sites, thus any algorithm able to compute the diagram of these sites allows to answer the predicate. This predicate is the basis of the randomized incremental construction algorithm for computing abstract Voronoi diagrams by Klein, Mehlhorn, and Meiser [16], as well as one of the main predicates analyzed by Emiris and Karavelas [12]. In [12] this predicate is decomposed to a number of subpredicates, one of them being the VConflict predicate.

We assume below that e lies on the bisector of S_i and S_j, oriented so that S_i is lying to the right of the bisector (refer also to Figure 11). The edge e inherits the orientation from its supporting bisector. The origin vertex of e is the Apollonius vertex defined by the (oriented) triple S_i, S_j and S_k, while the target vertex of e is the Apollonius vertex defined by the triple S_j, S_i and S_l. The EConflict predicate determines the type of the subset of e that is destroyed by the insertion of Q in the Apollonius diagram of the four sites, and has six possible outcomes:

- "conflict origin":

  a subsegment of e adjacent to its origin vertex disappears in the Apollonius diagram of the five sites. This case occurs if and only if $\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) =$ "conflict" and $\mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "no conflict". This case is illustrated by Q_{cn} in Figure 11.

- "conflict target": is the symmetric case that occurs iff $\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) =$ "no conflict" and $\mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "conflict". See Q_{nc} in Figure 11.

- "no conflict": no portion of e is destroyed by the insertion of Q in the Apollonius diagram of the four sites. This case can occur only when $\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "no conflict". See Q_{nn} in Figure 11.

- "conflict interior": a subsegment in the interior of e disappears in the Apollonius diagram of the five sites. This case can occur only when $\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "no conflict". See Q'_{nn} in Figure 11.

- "conflict entire edge":

  the entire edge e is destroyed by the addition of Q in the Apollonius diagram of the four sites. This case can occur only when $\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "conflict". See Q_{cc} in Figure 11.

- "conflict both": subsegments of e adjacent to its two vertices disappear in the Apollonius diagram of the five sites. This case can occur only when $\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "conflict". See Q'_{cc} in Figure 11.

Thus when the evaluations of predicate $\mathsf{VConflict}^{\varepsilon}$ on (S_i, S_j, S_k, Q) and (S_j, S_i, S_l, Q) are available, it only remains to distinguish between "no conflict" and "conflict interior", as well as between "conflict entire edge" and "conflict both". Assuming a non-degenerate configuration, this question is addressed in [12] using an auxiliary predicate of algebraic degree 16. The only situation where this auxiliary predicate has degeneracies is when $\mathsf{VConflict}(S_i, S_j, S_k, Q)$ or $\mathsf{VConflict}(S_j, S_i, S_l, Q)$ are "degenerate" and $\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q)$. Then the predicate can be answered by looking at the relative position of t_q with respect to t_i and t_j on ∂B_{ijk} (or the relative position of t'_q with respect to t'_i and t'_j on ∂B_{jil}).

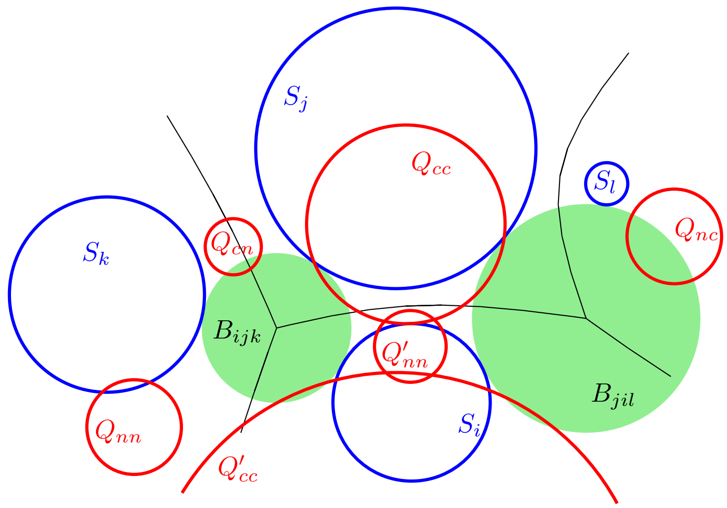


Figure 11: The different non-degenerate possible configurations for the EConflict predicate.

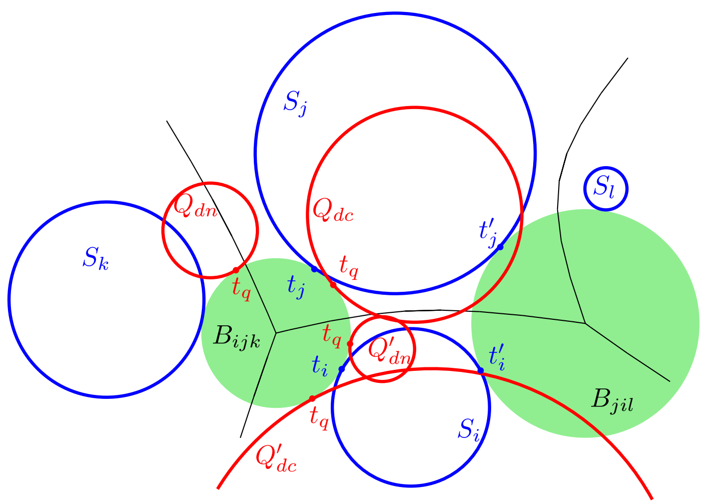


Figure 12: The different degenerate possible configurations for the EConflict predicate.

More precisely $\mathsf{EConflict}^{\varepsilon}(S_i, S_j, S_k, S_l, Q)$ can be evaluated by means of the following procedure. Figure 12 describes the different degeneracies that may occur.

1. **if** $(\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) =$ "conflict"
   and $\mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "no conflict") **then**
   return "conflict origin";

2. **if** $(\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) =$ "no conflict"

and $\mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "conflict") **then**
return "conflict target";

3. **if** ($\mathsf{VConflict}(S_i, S_j, S_k, Q) = \mathsf{VConflict}(S_j, S_i, S_l, Q)$
   and $\mathsf{VConflict}(S_i, S_j, S_k, Q) \neq$ "degenerate") **then**
   return $\mathsf{EConflict}(S_i, S_j, S_k, S_l, Q)$;

4. **if** $\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "no conflict" **then**
   if $\mathsf{VConflict}(S_i, S_j, S_k, Q) =$ "degenerate" **then**
   if $t_i t_j t_q$ is ccw **then return** "no conflict"; [Q_{dn} in Figure 12]
   else return "conflict interior"; [Q'_{dn} in Figure 12]
   else
   [in this case: $\mathsf{VConflict}(S_j, S_i, S_l, Q) =$ "degenerate"]
   if $t'_j t'_i t'_q$ is ccw **then return** "no conflict";
   else return "conflict interior";

5. **if** $\mathsf{VConflict}^{\varepsilon}(S_i, S_j, S_k, Q) = \mathsf{VConflict}^{\varepsilon}(S_j, S_i, S_l, Q) =$ "conflict" **then**
   if $\mathsf{VConflict}(S_i, S_j, S_k, Q) =$ "degenerate" **then**
   if $t_i t_j t_q$ is ccw **then return** "conflict both"; [Q_{dc} in Figure 12]
   else return "conflict entire edge"; [Q'_{dc} in Figure 12]
   else
   [in this case: $\mathsf{VConflict}(S_j, S_i, S_l, Q) =$ "degenerate"]
   if $t'_j t'_i t'_q$ is ccw **then return** "conflict both";
   else return"conflict entire edge";

The main observation from the above pseudocode is that the max-weight qualitative perturbation scheme described in Section 4.2 not only resolves the VConflict predicate, but also the EConflict predicate (although not described in this paper, our perturbation scheme resolves, in fact, all degeneracies of all predicates used in [12] for the computation of the 2D Apollonius diagram). To resolve the EConflict predicate, we need only evaluate $\mathsf{Orientation}(t_i, t_j, t_q)$ or $\mathsf{Orientation}(t'_j, t'_i, t'_q)$. As described in the previous section, this predicate is of algebraic degree 8, and thus does not increase the algebraic degree of the EConflict predicate. Furthermore, with a careful implementation, it is possible to keep track of the intermediate results of the evaluation of the $\mathsf{VConflict}^{\varepsilon}$ predicate and resolve the EConflict predicate using these intermediate results in a purely combinatorial manner.

As a final note, the above procedure for the evaluation of the EConflict predicate works as is even when the Apollonius edge e is of zero length. This is the case when the Apollonius vertices v_{ijk} and v_{jil} coincide, or, equivalently, when the four sites S_i, S_l, S_j and S_k are all tangent (and in that ccw order) to the same Apollonius circle. The only difference with respect to the non-zero-length Apollonius edge case are the possible outcomes: the EConflict predicate will never return "conflict interior" nor "conflict both"; this is, however, automatically handled by the procedure described above, that is without the need to handle any additional special cases.

4.4 Perturbing spheres for the 3D Apollonius diagram

In dimension three the most critical predicate is the VConflict predicate. Solving degeneracies for other predicates such as EConflict or FConflict boils down, as in dimension two, to solving VConflict degeneracies. An Apollonius vertex v_{ijkl} is defined by four sites S_i, S_j, S_k, and S_l, while the predicate $\mathsf{VConflict}(S_i, S_j, S_k, S_l, Q)$ tests if after adding a fifth site $Q = S_q$, vertex v_{ijkl} remains a valid Apollonius vertex or not. Let B_{ijkl} denote the open ball tangent to S_i, S_j, S_k, and S_l, whose tangency points t_i, t_j, t_k, and t_l form a positively oriented tetrahedron $t_it_jt_kt_l$.

The predicate in general position can be solved in different ways. One way is to do like Boissonnat and Delage [3]. Another way is to use inversion like in the 2D case by Emiris and Karavelas [12], and arrive at an alternative expression [13, 15]. In degenerate configuration, Q and B_{ijkl} are tangent at t_q and we can obtain, as a side product, the orientations of the tetrahedra formed by 4 of the 5 tangency points t_i, t_j, t_k, t_l, and t_q.

As for the 2D case, we apply the max-weight QSP scheme. If the predicate is degenerate the effect of the perturbation is that the weight of the site with largest weight increases and thus this site intersects the ball tangent to the four other sites. In the neighborhood of the center of B_{ijkl} the Apollonius diagram of S_i, S_j, S_k, S_l, and Q has the same combinatorial structure as the Voronoi diagram of t_i, t_j, t_k, t_l, and t_q. We, thus, get an equivalent formulation for the predicate: given five co-spherical points t_i, t_j, t_k, t_l, and t_q, does the tetrahedron $t_it_jt_kt_l$ remain in the Delaunay triangulation when the point of the largest index is moved inside the ball. Notice that it implies that the point with largest index is linked to all other points in this Delaunay triangulation.

Similarly to the two-dimensional case, we can conclude that Q is in conflict with B_{ijkl} if $q > i, j, k, l$. We can also take care of the cases where t_q is equal to one of the four points t_i, t_j, t_k, and t_l. Otherwise, we rename the indices so that i is the largest one. The definition of B_{ijkl} says that tetrahedron $t_it_jt_kt_l$ is positively oriented. Notice that this can be true in two ways: either the tetrahedron is really positively oriented, or it is flat, as the limit of a positively oriented tetrahedron when $\varepsilon_i \to 0^+$.

If $t_qt_jt_kt_l$ is positively oriented, t_q and t_i are on the same side of $t_jt_kt_l$, which is a facet of the convex hull of the five points. Since the 3D Apollonius graph is "star-shaped" from S_i, S_i is linked to $S_jS_kS_l$ to create the tetrahedron $S_iS_jS_kS_l$, and thus there is no conflict for Q.

If $t_qt_jt_kt_l$ is negatively oriented, t_q and t_i are on opposite sides of $t_jt_kt_l$, which implies that $S_jS_kS_l$ ceases to be a facet of the Apollonius graph. Thus, the tetrahedron $S_iS_jS_kS_l$ disappears and Q is in conflict.

If $t_qt_jt_kt_l$ is flat, the question reduces to determining the orientation of $t_q^\varepsilon t_j^\varepsilon t_k^\varepsilon t_l^\varepsilon$, which are the points of tangency of S_q, S_j, S_k, S_l with B_{ijkl}^ε after the perturbation of S_i by ε_i. This orientation will be non-degenerate except in two very special cases where the centers of Q, S_j, S_k, and S_l are either co-circular or collinear.

We first address the case where s_q, s_j, s_k, and s_l are neither co-circular nor collinear. Let us assume that l is smaller than j and k, which means that $w_l \leq w_j, w_k$. By subtracting w_l from all weights, we can consider that S_l has zero weight, and then perform an inversion with pole s_l (see Figure 13). Let Z_i, Z_j, Z_k, and Z_q be the images of sites S_i, S_j, S_k, and Q, and ω_i, ω_j, ω_k, and ω_q be the images of t_i, t_j, t_k, and t_q under inversion, and denote by z_i, z_j, z_k, and z_q the centers of Z_i, Z_j, Z_k, and Z_q. Since $t_qt_jt_kt_l$ is a flat tetrahedron,

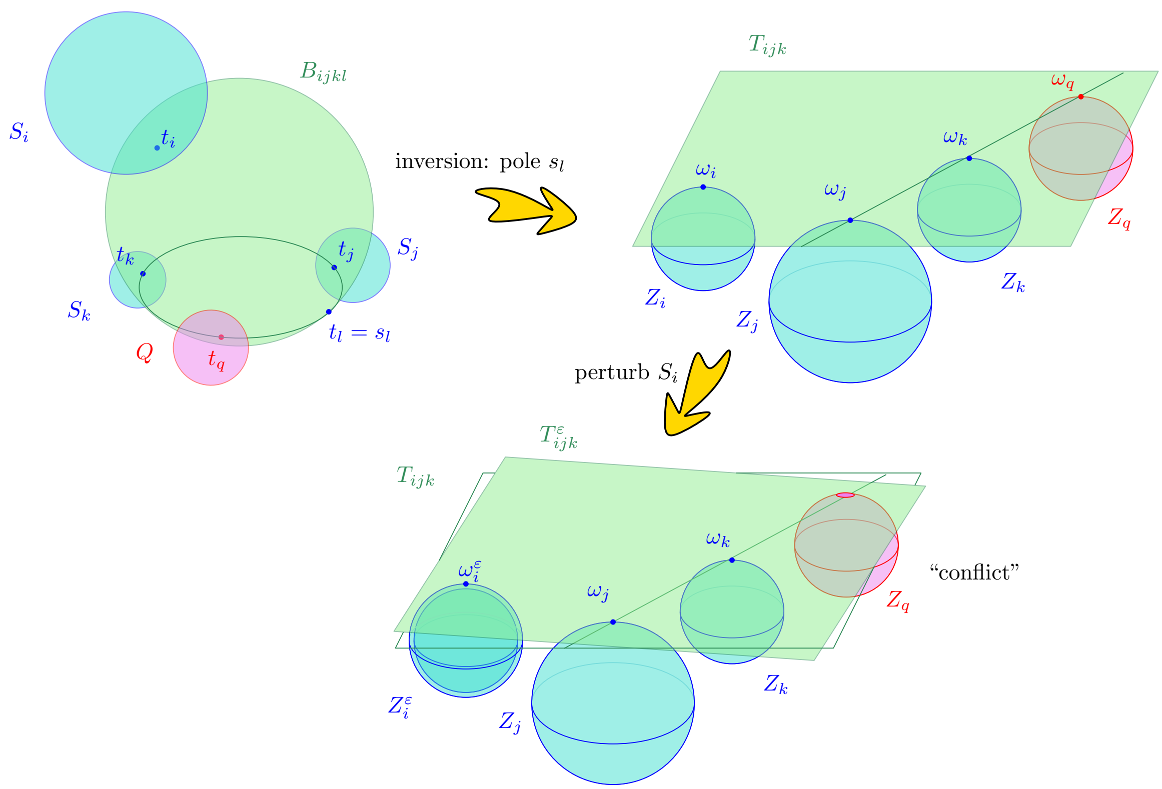


Figure 13: The case, for the VConflict predicate, where the tetrahedron $t_qt_jt_kt_l$ is flat.

the four points t_q, t_j, t_k, t_l are co-circular and thus $\omega_q, \omega_j, \omega_k$ are collinear. Their supporting line lies in the plane T_{ijk}, image of B_{ijkl} under inversion, and which contains ω_i, ω_j, and ω_k. The plane T_{ijk} is the *unique* plane commonly tangent to all three sites Z_q, Z_j, and Z_k. The uniqueness follows from the fact that z_q, z_j, and z_k are not collinear, since s_q, s_j, s_k, and $s_l = t_l$ have been assumed to be neither co-circular nor collinear. Let C denote the cone with axis z_jz_k, whose generatrices are common tangents to Z_j and Z_k, and whose apex lies outside the segment z_jz_k. Any plane tangent to both Z_j and Z_k and having Z_j and Z_k on the same side is also tangent to C. When we perturb S_i to S_i^{ε}, the plane T_{ijk} moves a bit, in the set of planes tangent to C, to become T^{ε}_{ijk}.

Consider first the case $w_q \geq w_l$, which implies that the weight of Z_q is non-negative. Since z_q, z_j, and z_k are not collinear and Z_q is tangent to C at ω_q, the sphere Z_q either properly intersects C or is inside C. If Z_q is (tangent to and) inside C, then, for all values of ε_i, the plane T^{ε}_{ijk} does not intersect Z_q, and the result of the perturbed predicate is "no conflict", otherwise, for all values of ε_i, the plane T^{ε}_{ijk} intersects Z_q, and the result of the perturbed predicate is "conflict". The way to evaluate the $\mathsf{VConflict}^{\varepsilon}$ predicate, in this case, is by determining the value of the orientation of $z_jz_kz_q$ in the plane that is perpendicular to T_{ijk} and passes through z_j and z_k, oriented such that $z_jz_k\omega_j$ is positive. If $w_j = w_k = w_l$, the cone C degenerates to the line through z_j and z_k. In this case we have $w_q > w_l$ (since, otherwise, s_j, s_k, s_l and s_q would have been co-circular), and thus $\mathsf{VConflict}^{\varepsilon}$ returns "conflict". If at least two of w_j, w_k, w_l differ, then at least one of ω_j and ω_k differs from z_j and z_k, respectively. Denoting by $\omega_\star$ a point of tangency that differs from the corresponding center, it suffices to determine if $\mathsf{Orientation}(z_q, z_j, z_k) = \mathsf{Orientation}(\omega_\star, z_j, z_k)$, in which case the $\mathsf{VConflict}^\varepsilon$ predicate returns "no conflict", otherwise "conflict" is returned.

Finally, notice that when $w_q < w_l$, the site Z_q has negative weight. In this case, the sphere Z_q^- will properly intersect the plane T_{ijk}^ε for all values of ε_i, which implies that S_q does not intersect B_{ijk}^ε. Hence, in this case, the result of the perturbed predicate is "no conflict".

In all cases above, perturbing S_i is sufficient to remove the degeneracy. In the very degenerate cases where s_j, s_k, s_l, and s_q are co-circular or collinear, the unique edge of the (degenerate) Apollonius diagram of S_j, S_k, S_l, and Q is a circle or a line. In these cases, the position of S_i has no influence on the combinatorial structure of the diagram of S_i, S_j, S_k, S_l, and Q, and we need to perturb the second most perturbed site S_j or S_k or Q to remove the degeneracy. The resolution of the degeneracy is similar to the 2D case: first perform a positive permutation of j, k, l to ensure that $j > k, l$. If $q > j$ then Q will be in conflict, otherwise, if $q < j$ then Q will be in conflict if and only if $t_j t_k t_l$ and $t_q t_k t_l$ have different two-dimensional orientations.

Following the above analysis, $\mathsf{VConflict}^\varepsilon(S_i, S_j, S_k, S_l, Q)$ can be evaluated as follows:

1. **if** $\mathsf{VConflict}(S_i, S_j, S_k, S_l, Q) \neq$ "degenerate" **then return** $\mathsf{VConflict}(S_i, S_j, S_k, S_l, Q)$;
2. **if** $q > \max\{i, j, k, l\}$ **then return** "conflict";
3. ensure that $i > \max\{j, k\} \geq \min\{j, k\} > l$ by a positive permutation of (i, j, k, l);
4. **if** $t_q = t_i$ **then return** "no conflict";
5. **if** $t_q = t_j$ **then** { **if** $q > j$ **then return** "conflict"; **else return** "no conflict"; };
6. **if** $t_q = t_k$ **then** { **if** $q > k$ **then return** "conflict"; **else return** "no conflict"; };
7. **if** $t_q = t_l$ **then** { **if** $q > l$ **then return** "conflict"; **else return** "no conflict"; };
8. **if** $t_q t_j t_k t_l$ is positively oriented **then return** "no conflict";
9. **if** $t_q t_j t_k t_l$ is negatively oriented **then return** "conflict";
10. **if** s_j, s_k, s_l, s_q are neither collinear nor co-circular **then**
    i. **if** $w_q < w_l$ **then return** "no conflict";
    ii. **if** $w_j = w_k = w_l$ **then return** "conflict";
    iii. compute a tangency point $\omega_\star$;
    if z_q, z_j, z_k and $\omega_\star, z_j, z_k$ have the same 2D orientation **then return** "no conflict";
    else return "conflict";
11. ensure that $i > j > \max\{k, l\}$ by a positive permutation of (j, k, l);
12. **if** $q > j$ **then return** "conflict";

13. **if** $t_jt_kt_l$ and $t_qt_kt_l$ have the same 2D orientation **then return** "no conflict";
else return "conflict";

Steps 8 and 9 of the above algorithm rely on the auxiliary predicate $\mathsf{Orientation}(t_q, t_j, t_k, t_l)$ that, given five sites, computes the orientation of the four tangency points on the common tangent sphere to the fifth site. Since the tetrahedron $t_it_jt_kt_l$ is, by definition positively oriented, $\mathsf{Orientation}(t_q, t_j, t_k, t_l)$ will be positive if and only if t_q lies on the same *half* of B_{ijkl} as t_i, where the two halves of B_{ijkl} are delimited by the circle through t_j, t_k and t_l. We first reduce all the weights by w_l. If $w_j = w_k = w_l (= 0)$, computing the orientation of t_q, t_j, t_k, t_l amounts to evaluating the orientation of t_q, s_j, s_k, s_l. If w_j, w_k and w_l are not all equal, we consider again the inversion transformation with s_l as the pole. Then t_q lies on the same half of B_{ijkl} as t_i if and only if ω_q lies on the same half-plane of T_{ijk}, with respect to the line through ω_j and ω_k, as ω_i. The equality of these 2D orientation tests is equivalent to testing the result of $\mathsf{Orientation}(z_q, z_j, z_k, \omega_\star)$; if $\mathsf{Orientation}(z_q, z_j, z_k, \omega_\star) =$ "ccw" return "no conflict", otherwise return "conflict".

For Step 10 we first need to test if the points s_j, s_k, s_l and s_q are either collinear or co-circular. The possible collinearity can easily be tested via the cross-products $(s_l - s_j) \times (s_k - s_j)$ and $(s_q - s_j) \times (s_k - s_j)$; if both are the zero vector, the four points are collinear. Co-circularity in the original space corresponds to collinearity in the inverted space (where the pole of inversion is s_l); to test if s_j, s_k, s_l and s_q are co-circular it suffices to test if z_j, z_k, z_q are collinear, which amounts to computing the cross-product $(z_q - z_j) \times (z_k - z_j)$. The 2D orientations of z_q, z_j, z_k and $\omega_\star, z_j, z_k$ are evaluated by determining, for a point $x \notin plane(z_qz_jz_k)$, if the two tetrahedra $z_qz_jz_kx$ and $\omega_\star z_jz_kx$ have the same orientation.

Finally, if the centers s_j, s_k, s_l, s_q are co-circular the 2D orientations of $t_jt_kt_l$ and $t_qt_kt_l$ in Step 13 are the same as those of $s_js_ks_l$ and $s_qs_ks_l$. Choosing a point $x \notin plane(s_js_ks_l)$, we may compute these 2D orientations via their 3D counterparts $s_js_ks_lx$ and $s_qs_ks_lx$. If the centers s_j, s_k, s_l, s_q are collinear, notice that, due to the fact that the tetrahedron $t_it_jt_kt_l$ is positively oriented, s_k lies inside the segment s_js_l. Hence, determining if the 2D orientations of $t_jt_kt_l$ and $t_qt_kt_l$ are the same reduces to determining if s_q lies inside the segment s_ks_l or not; in the former case we return "conflict", while in the latter case we return "no conflict".

We end this section by briefly discussing the algebraic degree of three-dimensional $\mathsf{VConflict}^\varepsilon$ predicate. The unperturbed predicate can be evaluated with algebraic expressions of degree at most 10 [13]; this accounts for Step 1 of the algorithm described above. Steps 2, 3, 11 and 12 all amount to comparing indices of sites, so they are all of degree 1. To resolve Steps 4 to 7 we need to test whether the points of tangency of two spheres with the common Voronoi sphere coincide; this amounts to testing whether one sphere is internally tangent to another one, which is a degree-2 predicate. Step 13 can easily be resolved using expressions of degree at most 6 [15], while Steps 10(i) and 10(ii) are clearly degree-1 operations. The most demanding parts of our evaluation procedure are Steps 8, 9 and 10(iii). Their algebraic degrees can be shown to be 28 and 20, respectively [15]. Hence, the $\mathsf{VConflict}^\varepsilon$ predicate can be evaluated, as described above, with expressions of algebraic degree at most 28.

5 Conclusion

In this paper, a new framework for dealing with geometric degeneracies has been proposed: QSP. Conversely to usual approaches for symbolic perturbation, the new framework does not rely on a particular algebraic description of the predicate, but rather directly on its geometric description.

A QSP scheme consists of a sequence of perturbations, but given a specific predicate only a few of these perturbations are really *active*. The number of active perturbations used to resolve a specific predicate depends on the problem at hand. For the 2D Apollonius diagram perturbing one site always suffices. In its 3D counterpart we may need to perturb two sites, whereas in the case of circular arcs we may need perform a rotation (perturb the axes) and perturb up to one supporting circle per predicate. Minimizing the number of active perturbations is not necessarily desirable, since it might result in a more complicated design for the perturbed predicate (for example, trying to resolve degeneracies for the trapezoidal map of circular arcs with a single active perturbation seems much more complicated).

Besides the number of active perturbations, another important issue is the ordering of the perturbations: for the Apollonius diagram we consider sites by decreasing weight, whereas for the arrangement of circular arcs we first consider a (global) rotation and then the circles by means of decreasing radius. Different perturbation sequences than the ones described in this paper are definitely possible; the analysis, however, can become unnecessarily more complicated.

Our qualitative symbolic perturbation framework, and in particular the schemes described in this paper, can also be applied to a variety of other problems, such as the 2D Voronoi diagram of disjoint convex objects under any L_p metric, as well as the Euclidean Voronoi diagram of certain disjoint convex objects in 3D (the objects can be, for example, non-intersecting lines, line segments or rays). It suffices to replace a site S_i by its Minkowski sum with a ball of radius ε_i, and then consider the limits $\varepsilon_i \to 0^+$, for an appropriately defined ordering of the sites. Another type of geometric problem, involving complex predicates, for which the QSP framework is relevant, is the computation of lines tangent to four given lines in 3D [4, 7].

Acknowledgments The authors warmly thank the reviewers for their very careful reading of the paper and their detailed comments and suggestions.

References

[1] Pierre Alliez, Olivier Devillers, and Jack Snoeyink. Removing degeneracies by perturbing the problem or the world. *Reliable Computing*, 6:61–79, 2000. URL: http://hal.inria.fr/inria-00338566/, doi:10.1023/A:1009942427413.

[2] J.-D. Boissonnat and J. Snoeyink. Efficient algorithms for line and curve segment intersection using restricted predicates. *Comput. Geom. Theory Appl.*, 16(1):35–52, 2000. doi:10.1016/s0925-7721(99)00057-7.

[3] Jean-Daniel Boissonnat and Christophe Delage. Convex hull and Voronoi diagram of additively weighted points. In G. S. Brodal and S. Leonardi, editors, *Proceedings of 13th Annual European Symposium on Algorithms (ESA 2005)*, volume 3669 of *LNCS*, pages 367–378. Springer, 2005. URL: http://www.springerlink.com/index/ehwb4lu0h7kklj61.pdf.

[4] H. Brönnimann, O. Devillers, Vida Dujmović, H. Everett, M. Glisse, X. Goaoc, S. Lazard, H.-S. Na, and S. Whitesides. Lines and free line segments tangent to arbitrary three-dimensional convex polyhedra. *SIAM Journal on Computing*, 37:522–551, 2007. URL: http://hal.inria.fr/inria-00103916, doi:10.1137/S0097539705447116.

[5] C. Burnikel, K. Mehlhorn, and S. Schirra. On degeneracy in geometric computations. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 16–23, 1994.

[6] Olivier Devillers, Alexandra Fronville, Bernard Mourrain, and Monique Teillaud. Algebraic methods and arithmetic filtering for exact predicates on circle arcs. *Comput. Geom. Theory Appl.*, 22:119–142, 2002. URL: https://hal.inria.fr/inria-00166709, doi:10.1016/S0925-7721(01)00050-5.

[7] Olivier Devillers, Marc Glisse, and Sylvain Lazard. Predicates for line transversals to lines and line segments in three-dimensional space. In *Proc. 24th Annual Symposium on Computational Geometry*, pages 174–181, 2008. URL: http://hal.inria.fr/inria-00336256/, doi:10.1145/1377676.1377704.

[8] Olivier Devillers, Menelaos I. Karavelas, and Monique Teillaud. Qualitative Symbolic Perturbation: a new geometry-based perturbation framework. Research Report 8153, INRIA, 2012. URL: http://hal.inria.fr/hal-00758631/.

[9] Olivier Devillers and Monique Teillaud. Perturbations for Delaunay and weighted Delaunay 3D triangulations. *Computational Geometry: Theory and Applications*, 44:160–168, 2011. URL: http://hal.archives-ouvertes.fr/inria-00560388/, doi:10.1016/j.comgeo.2010.09.010.

[10] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9(1):66–104, 1990. doi:10.1145/77635.77639.

[11] I. Emiris and J. Canny. A general approach to removing degeneracies. *SIAM J. Comput.*, 24:650–664, 1995. doi:10.1137/S0097539792235918.

[12] Ioannis Z. Emiris and Menelaos I. Karavelas. The predicates of the Apollonius diagram: algorithmic analysis and implementation. *Computational Geometry: Theory and Applications*, 33(1-2):18–57, January 2006. Special Issue on Robust Geometric Algorithms and their Implementations. doi:10.1016/j.comgeo.2004.02.006.

[13] Iordan Marinov Iordanov. The Euclidean *InSphere* Predicate. B.S. thesis, University of Crete, Department of Applied Mathematics, 2013. URL: http://www.tem.uoc.gr/~mkaravel/files/theses/Iordanov-BSthesis.pdf.

[14] Geoffrey Irving and Forrest Green. A deterministic pseudorandom perturbation scheme for arbitrary polynomial predicates. Technical Report 1308.1986v1, arXiv, 2013. URL: http://arxiv.org/abs/1308.1986.

[15] Emmanouil Kamarianakis. Personal communication, 2013.

[16] Rolf Klein, Kurt Mehlhorn, and Stefan Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Comput. Geom. Theory Appl.*, 3(3):157–184, 1993. doi:10.1016/0925-7721(93)90033-3.

[17] Giuseppe Liotta, Franco P Preparata, and Roberto Tamassia. Robust proximity queries: An illustration of degree-driven algorithm design. *SIAM Journal on Computing*, 28(3):864–889, 1998. doi:10.1137/S0097539796305365.

[18] K. Mehlhorn, R. Osbild, and M. Sagraloff. A general approach to the analysis of controlled perturbation algorithms. *Comput. Geom. Theory Appl.*, 44:507–528, 2011. doi:10.1016/j.comgeo.2011.06.001.

[19] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.*, 19:1–17, 1998. doi:10.1007/PL00009330.

[20] Raimund Seidel. Perturbations in geometric computing, 2013. Talk at the Workshop on Geometric Computing, Heraklion. URL: http://www.acmac.uoc.gr/GC2013/files/Seidel-slides.pdf.

[21] C. K. Yap. A geometric consistency theorem for a symbolic perturbation scheme. *J. Comput. Syst. Sci.*, 40(1):2–18, 1990. doi:10.1016/0022-0000(90)90016-E.

[22] C. K. Yap. Symbolic treatment of geometric degeneracies. *J. Symbolic Comput.*, 10:349–370, 1990. URL: http://www.sciencedirect.com/science/article/pii/S0747717108800697.

[23] C. K. Yap and T. Dubé. The exact computation paradigm. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, volume 4 of *Lecture Notes Series on Computing*, pages 452–492. World Scientific, Singapore, 2nd edition, 1995. URL: http://www.cs.nyu.edu/~exact/doc/paradigm.ps.gz.